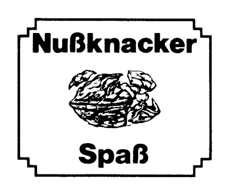
Neue Spiele und Programme zum Commodore 64



Neue Spiele und Programme zum Commodore 64



Neue Spiele und Programme zum Commodore 64

herausgegeben von

Steffi Siol



Verlag Rudolf Müller · Köln

ISBN 3-481-36321-4

Illustration: Typo-Studio Frey, Düsseldorf
Titelgestaltung: Werbestudio Schulz, Köln
Satz: Fotosatz Böhm, Köln
Druck: DVG, Darmstadt
Printed in Germany

Inhalt

Viel Spaß mit dem Commodore 64



Im Verhältnis zu seinem doch noch recht günstigen Anschaffungspreis bietet der Commodore 64 unheimlich viele Möglichkeiten. Besonders die professionelle Tastatur hebt ihn von anderen Rechnern seiner Klasse ab.

Eine interessante Zusammenstellung

Wir haben Ihnen in diesem Buch eine Reihe von Programmen zusammengestellt, mit denen Sie auf diesem Rechner sicher viel Spaß haben werden.

Verständliche Texte und Programme

Besonderen Wert haben wir auf die Verständlichkeit der Texte und Programme gelegt. Die sehr ausführlichen Erklärungen sollen Ihnen helfen, die Programme wirklich zu verstehen, um so im Programmieren sicherer zu werden.

Häufig enthalten andere Programme unheimlich viele verschlüsselte Informationen, die dann auch im Text nicht erklärt sind. Besonders die Eingabe von Maschinenprogrammen aus Basic ist sehr beliebt und —nach unserer Meinung— wenig schön. Wir haben darauf völlig verzichtet.

Spaß gehört immer dazu

Programme sollen verständlich und übersichtlich sein, und es soll Spaß machen, sie zu benutzen. Das waren unsere Kriterien beim Zusammenstellen dieses Buches.

Wir haben uns darüber hinaus bemüht, in jedem Programm eine Programmierstrategie oder einige Sprachelemente besonders zu verdeutlichen. Davon können Sie profitieren.

Mit dem Grundgerät getestet

Ein Wort noch zur von uns benutzten Hardware. Alle Programme sind auf dem Grundgerät getestet. Es sind keinerlei Erweiterungen oder zusätzliche Geräte notwendig. Denn mit der Grundversion kann man schon unheimlich viel machen.

Legen Sie nun also los. Ihr Computer wartet schon ungeduldig auf die neuen Programme.



Dieses Programm erlaubt es Ihnen, mit dem Computer ein persönliches Gespräch über Ihre Daten zu führen. Das Gespräch ist recht kurz, aber es liegt an Ihnen, es zu verlängern.

Unterhaltungsprogramm mit Gewinn

Das wichtigste Element eines solchen Unterhaltungsprogramms ist die Analyse der möglichen Antworten.



Bevor wir in die Beschreibung des Programms einsteigen, wollen wir Ihnen einen Vorschlag machen. Solche Programme kann man beinahe endlos lang machen. Dazu braucht man nur gute Ideen.

Verlängerung mit Prämie

„Bauen" Sie eine Verlängerung für das Programm und schicken es uns unter dem Stichwort „Unterhaltungsprogramm". Die besten Lösungen werden prämiert und falls wir Ihr Programm veröffentlichen, erhalten Sie zusätzlich ein Honorar.

Vergessen Sie auch Ihren Namen und Ihre Anschrift nicht, sonst wissen wir gar nicht, wohin wir den Preis schicken sollen.

Nur wenig ausgewertet

Nun zum Programm: Am Anfang fragt Sie der Computer einfach nur, ob Sie sich mit ihm unterhalten wollen. Ausgewertet wird nur, ob die Antwort „NEIN" ist.

Machen Sie es besser und schöner

Da bieten sich natürlich noch viele andere Möglichkeiten. Lassen Sie den Computer auf unterschiedliche Antworten ganz unterschiedlich reagieren, und denken Sie sich eine schöne Variante für die Möglichkeit aus, daß jemand eine Antwort gibt, die Sie nicht vorhergesehen haben. Im zweiten Teil des Programms werden Ihre persönlichen Daten ab-

gefragt. Die Eingaben werden jeweils bei den nächsten Fragen verwendet.

Der Computer analysiert Ihr Alter

Ausführlich wird im dritten Teil das Alter ausgewertet. Da bekommen Sie schon ganz individuelle Kommentare.

Natürlich kann dieses kurze Programm nur ein Beispiel und Anregung sein. Lassen Sie Ihre Phantasie spielen und gewinnen Sie einen der attraktiven Preise.

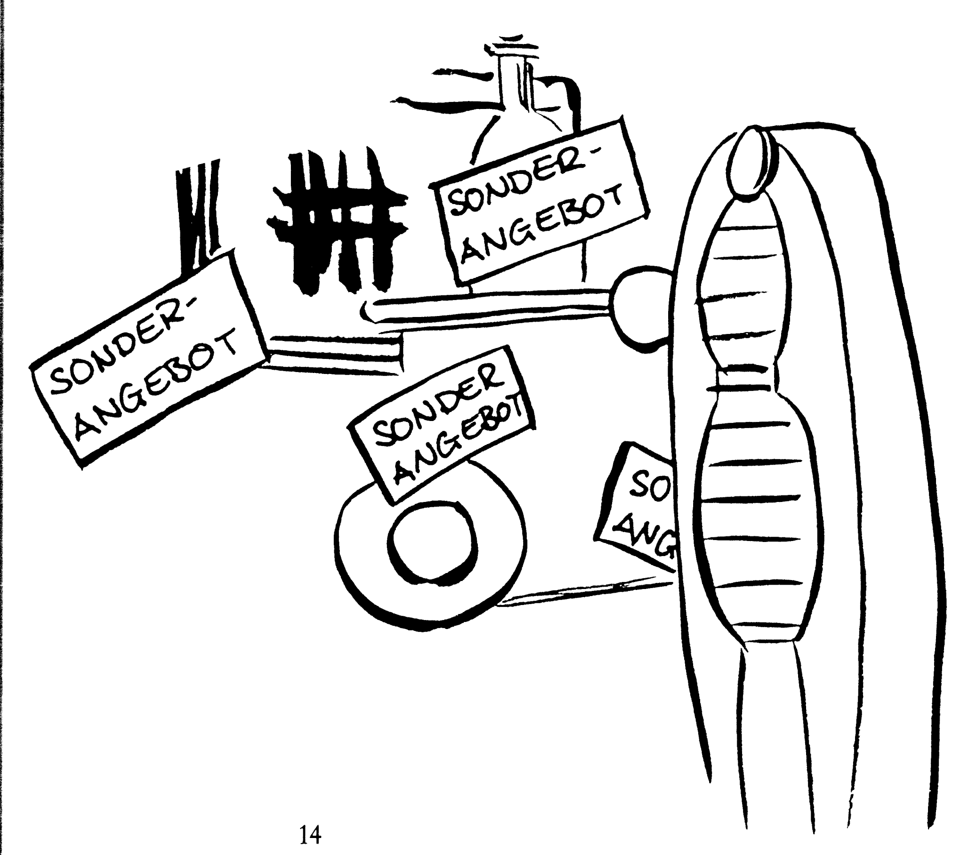


```
110 REM ********************************************************
120 REM *  Nußknacker Programm   Personendaten  (44)           *
130 REM *                                                      *
140 REM *  1. Teil   -   Basisdaten   -   Initialisierung      *
150 REM ********************************************************
160 PRINT "HALLO, HIER IST DAS"
170 PRINT "NUSSKNACKER PROGRAMM"
180 PRINT "FUER EIN PERSOENLICHES"
190 PRINT "GESPRAECH UEBER IHRE"
200 PRINT
210 PRINT "*****  DATEN *****"
220 PRINT
230 PRINT "WOLLEN SIE SICH MIT MIR"
240 PRINT "UNTERHALTEN (JA/NEIN)?"
250 INPUT A$
260 IF A$="NEIN" THEN GOTO 520
270 REM ********************************************************
280 REM *  2. Teil   -   Abfrage                               *
290 REM ********************************************************
300 DIM P$(3)
310 PRINT "WIE IST IHR NACHNAME ?"
320 INPUT P$(1)
330 PRINT P$(1)
340 PRINT "EIN NETTER NAME"
350 PRINT "UND WIE IST IHR VORNAME ?
360 INPUT P$(2)
370 PRINT "HALLO, ";P$(2);" ";P$(1)
380 PRINT "IN WELCHEM JAHR SIND SIE GEBOREN ?"
390 INPUT P
400 LET X=1984-P
410 REM ********************************************************
420 REM *  3. Teil   -   Auswertung
430 REM ********************************************************
440 IF X<1 OR X>100 THEN PRINT "LUEGNER"
450 IF X>0 AND X<6 THEN PRINT "HALLO, BABY"
460 IF X>5 AND X<17 THEN PRINT P;" - WIRKLICH SCHOENES ALTER"
470 IF X=17 THEN PRINT "MIT 17 HAT MAN NOCH TRAEUME ..."
480 IF X>17 AND X<25 THEN PRINT "DIE ERSTE LIEBE SCHON VORBEI"
490 IF X>24 AND X<55 THEN PRINT "IMMEN SCHOEN ARBEITEN"
500 IF X>55 AND X<75 THEN PRINT "ES LEBE DIE RENTE"
510 IF X>74 THEN PRINT "ALLE ACHTUNG"
520 PRINT
530 PRINT "AUF WIEDERSEHEN  "
540 REM ********************************************************
```

Vielleicht haben Sie auch schon öfter das Problem gehabt, daß Sie in der Bank zwar Geld tauschen konnten, die Umrechnungstabelle aber auf einem so alten Kurs basierte, daß sie unbrauchbar war.

Aktuelle Umrechnungstabelle

Mit diesem Programm können Sie sich mit dem ganz aktuellen Kurs Ihre eigene Umrechnungstabelle für Ihren Urlaub ausrechnen lassen.



Das Programm ist so aufgebaut, daß es eine zweiteilige Tabelle ausgibt. Im ersten Teil der Tabelle ist die Basis die eigene Währung, also die DM.

In diesem Teil der Tabelle können Sie nachsehen, wieviel Sie von der ausländischen Währung für Ihr Geld bekommen.

Der zweite Teil der Tabelle ist genau umgekehrt. Wenn Ihnen unterwegs irgendwo im Laden ein „besonderes Angebot" in's Auge fällt, oder wenn der Hotelportier Ihnen den Zimmerpreis nennt, können Sie in diesem Teil der Tabelle ganz schnell feststellen, wieviel DM Sie auf den Tisch legen müßten.

Mögen Sie lange Tabellen?

Wir haben in diesem Programm eine ziemlich ausführliche Tabelle vorgeschlagen. Jeder Teil der Tabelle hat 100 Positionen.

Sie können ablesen, wieviel von der ausländischen Währung Sie für eine DM bekommen. Sie können aber auch ablesen, was 100 DM in der ausländischen Währung wert sind. Und dazwischen ist jede volle DM aufgeführt.

Wie teuer ist das in DM?

Genauso ist es umgekehrt im zweiten Teil der Tabelle. Dort können Sie die zugehörigen DM-Werte von einer bis zu 100 Einheiten der fremden Währung ablesen.

Das mögen Ihnen zu viele Zahlen sein, oder Sie möchten in Ihrer Tabelle gern höhere Werte sehen. Wir erklären Ihnen gleich, wo Sie im Programm Änderungen vornehmen müssen.

Nach der Begrüßung fragt das Programm im zweiten Teil zuerst nach Ihrem Reiseziel. Ihre Eingabe wird eingelesen und als Zeichenfolge unter L$ (wie Land) gespeichert. Daraus entsteht später die Überschrift für Ihre Tabelle.

In welcher Währung wollen Sie rechnen?

Natürlich muß über der Tabelle auch der Name der Währung stehen. Die fragt das Programm anschließend ab und speichert Ihre Eingabe unter W$ (wie Währung).

Dann fehlt für die Berechnung nur noch der aktuelle Kurs. Den erfahren Sie bei der Bank oder aus der Zeitung. Gespeichert wird der Kurs unter K (wie Kurs). Natürlich hat das K kein Dollarzeichen. Hier wird ja eine Zahl gespeichert. Im Gegensatz dazu waren die ersten beiden Eingaben Zeichenfolgen (gekennzeichnet durch $).

Land, Kurs, Währung und Tabelle

Jetzt berechnet der Computer den ersten Teil der Tabelle und gibt die Werte gleich aus, nachdem er die Überschrift mit dem Land (L$), dem Kurs (K) und der Währung (W$) ausgegeben hat.

Beachten Sie in Zeile 400 das Komma zwischen der Zeichenfolge „DM“ und der Variablen für die Währung W$. Damit sorgt der Computer automatisch für Platz zwischen den beiden Texten.

In Zeile 420 beginnt dann die Schleife für die Berechnung und Ausgabe der Fremdwäh-

rung im Verhältnis zu glatten DM-Beträgen zwischen 1 und 100. An dieser Stelle können Sie auch Art und Umfang der Tabelle verändern.

Ändern Sie die Tabelle nach Bedarf

Der linke Wert vor dem TO gibt den Anfangswert der Tabelle an. Er ist in unserem Beispiel auf 1 gesetzt. Setzen Sie ihn beliebig höher, wenn Sie eher an großen Beträgen interessiert sind. Rechts von TO steht der höchste DM-Betrag der Tabelle. Auch den können Sie nach oben oder nach unten verändern.

Falls Ihnen der Abstand zwischen den einzelnen Werten in der Tabelle nicht gefällt, können Sie mit STEP und einem angehängten Zahlenwert auch noch die Schrittweite verändern. Vielleicht interessiert Sie nur, was alle Vielfachen von fünf DM wert sind, vielleicht geben Sie sich mit einzelnen Markstücken gar nicht ab.

Das Rechnen fällt dem Computer leicht

Die eigentliche Berechnung ist sehr einfach: Der DM-Betrag wird einfach durch den Kurs geteilt.

Die zweite Schleife, die bei 520 beginnt, funktioniert ganz ähnlich. Nur entspricht die Laufvariable I hier einer Einheit der Fremdwährung, nicht einer DM wie in der ersten Schleife.

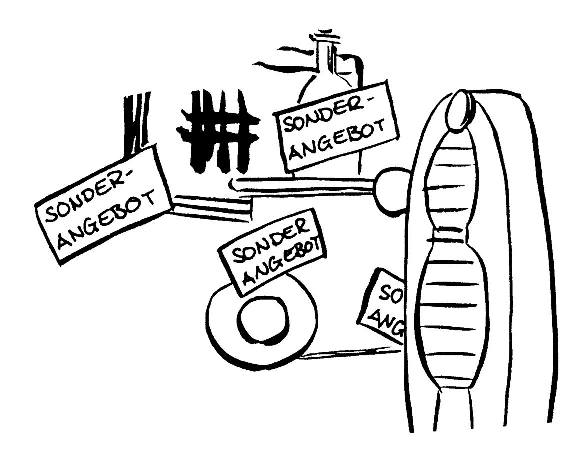
Entsprechend können Sie hier die Gestalt der Tabelle durch Veränderung von Anfangs- und Endwert sowie durch Angabe der Schrittweite mit STEP verändern.

Bei der japanischen Einheit Yen ist es beispielsweise ziemlich uninteressant, wieviel ein Yen wert ist. Interessanter sind schon 100 Yen, und 100 wäre auch die richtige Schrittweite.

Wir wünschen gute Reise

Die Berechnung in Zeile 530 läuft natürlich auch umgekehrt zu der in Zeile 430. Jetzt ist Mulitplikation gefragt.

Sie sehen, mit diesem Programm können Sie noch eine ganze Menge mehr machen, als es auf den ersten Blick scheint. Wir wünschen Ihnen viel Spaß beim Programmieren und Rechnen und natürlich eine gute Reise.



```
110 REM *******************************************************
120 REM *  Nußknacker Programm   Währungstabelle     (413)    *
130 REM *                                                     *
140 REM *  1. Teil   -   Begrüßung                            *
150 REM *******************************************************
160 PRINT "HALLO, HIER IST EIN"
170 PRINT "NUSSKNACKER PROGRAMM"
180 PRINT
190 PRINT "ES DRUCKT IHNEN EINE TABELLE"
200 PRINT "FUER IHRE URLAUBSWAEHRUNG"
210 PRINT
220 REM *******************************************************
230 REM *  2. Teil   -  Eingabe der Basisdaten                *
240 REM *******************************************************
250 PRINT "IN WELCHES LAND REISEN SIE?"
260 INPUT L$
270 PRINT "WIE HEISST DIE URLAUBSWAEHRUNG?"
280 INPUT W$
290 PRINT "WIE IST DER KURS?"
300 PRINT "EIN ";W$;" IST WIEVIEL DM?"
310 INPUT K
340 REM *******************************************************
342 OPEN 1,4
350 PRINT#1,"UMRECHNUNGSTABELLE FUER ";L$
360 PRINT#1,"TEIL 1"
370 PRINT#1,
380 LPRINT "KURS: ";K
390 PRINT#1,
400 PRINT#1,"DM",W$
410 PRINT#1,
420 FOR I = 1 TO 100
430 PRINT#1,I,I/K
440 NEXT I
450 PRINT#1,
460 PRINT#1,"TEIL 2"
470 PRINT#1,
480 PRINT#1,"KURS: ";K
490 PRINT#1,
500 PRINT#1,W$,"DM"
510 PRINT#1,
520 FOR I = 1 TO 100
530 PRINT#1,I,I*K
540 NEXT I
545 CLOSE 1
550 REM *******************************************************
```

Mit diesem Programm können Sie einen Brief eingeben und ihn anschließend an viele Leute verschicken, ohne ihn jedesmal neu abzutippen. Sie ergänzen einfach die Adresse.

Ein Brief für viele Leute

Damit wird Ihr Heimcomputer beinahe zu einem richtigen Textsystem. Dieses Programm können Sie um viele Feinheiten ergänzen.



Vorgesehen ist in Ihrem Programm Platz für einen Brief von maximal 30 Zeilen. Den Platz im Speicher reserviert die Anweisung 260. Der Text ist unter B$ (Brief) gespeichert.

Zusätzlich wird in der Anweisung 250 Platz für die Anschrift reserviert. Die Anschrift (A$) können Sie später immer wieder ändern, ohne den eigentlichen Text des Briefes zu ändern. Sie können so den gleichen Brief an viele verschiedene Anschriften senden. Serienbrief nennt man so was.

Geben Sie den Brieftext ein

In Zeile 460 beginnt dann die Eingabeschleife für den Brieftext. Falls Sie längere Briefe schreiben wollen — vorausgesetzt Ihr Computer verfügt über ausreichenden Speicherraum — können Sie den Wert hinter TO höher setzen. Sie müssen dann allerdings auch die 30 in der DIM-Anweisung 260 höher setzen.

Zeilenlänge überprüfen

Eine Zeile wird vorläufig unter dem Namen Z$ (Zeile) untergebracht, die das Programm in Zeile 480 von der Tastatur anfordert.

Mehr als 60 Zeichen gehören nicht in eine Briefzeile. Das kontrolliert das Programm in Zeile 500 und erteilt Ihnen in Zeile 550 einen Verweis, falls Sie eine längere Zeile eingegeben haben. Wenn Sie anderer Meinung sind, müssen Sie die 60 in Zeile 500 entsprechend verändern.

Leerzeilen sind selbstverständlich erlaubt,

und entsprechend fragt das Programm auch keine untere Grenze der Länge der Zeile (LEN) ab.

Ist Ihre Zeile dann für gut befunden, wird sie endgültig unter B$ abgespeichert (Zeile 510).

30 Zeilen oder Stern

Wenn Ihr Text genau 30 Zeilen hat, kommen Sie von 530 automatisch in Zeile 612. Sie können aber auch weniger Zeilen schreiben und teilen dem Computer durch eine Zeile, die nur einen Stern enthält, mit, daß Sie genug geschrieben haben. Sie landen dann von Zeile 490 ebenfalls in 612. Der Stern wird natürlich nicht gespeichert.

Die Anschrift nicht vergessen



Sie können dann in der Schleife ab Zeile 664 die erste Anschrift angeben, an die der Brief geschickt werden soll. Vorgesehen sind maximal sechs Zeilen.

Wollen Sie längere Anschriften, ändern Sie die 6 in Zeile 664 entsprechend. Sie müssen dann aber auch in Anweisung 250 mehr Speicher reservieren.

Natürlich gibt die 6 als obere Schleifenbegrenzung in 664 nur einen Maximalwert an. Genau wie beim Brieftext können Sie die Eingabe durch ein Sternchen für beendet erklären. Das Programm springt dann in Zeile 706.

Die maximal vorgesehene Länge für eine Adreßzeile ist 40. Geprüft wird das in Zeile 674. Längere Zeilen akzeptiert das System

nur, wenn Sie hier statt der 40 einen höheren Wert einsetzen.

Leerzeilen sind natürlich auch im Adreßteil erlaubt. Eine gehört ja mindestens zwischen Straße und Ort.

Wenn die sechs Zeilen komplett sind, springt das Programm aus Zeile 682 in Zeile 706. Hier landen Sie in jedem Fall, egal ob Sie sechs Zeilen eintippen oder früher mit einem Sternchen die Eingabe beenden.

Nun auf's Papier mit dem Brief

Da beginnt dann auch das Drucken des Briefes. Zuerst kommt ab Zeile 724 die Anschrift an die Reihe. Hier taucht wieder die 6 auf für die sechs möglichen Zeilen.

Wenn Sie mehrere Adressen hintereinander angeben und die Eingabe mit Stern beenden, kann es Ihnen passieren, daß unter A$ noch ein Rest der vorigen Adresse gespeichert ist. Sie sollten daher immer komplette sechs Zeilen eingeben. Ist die Adresse kürzer, geben Sie einfach Leerzeilen ein.

Restzeilen unterdrücken

Um dies zu vermeiden, können Sie aber auch in Zeile 672 zu einer Anweisungsfolge springen, die die restlichen Zeilen nach Angabe eines Sternchens löscht. Eine solche Anweisungsfolge müßte logisch nach der Zeile 694 stehen.

In der jetzigen Programmversion folgen auf das Drucken der Anschrift einige Leerzeilen und dann in der Schleife ab Zeile 754 der

eigentliche Text. Ausgedruckt werden immer 30 Zeilen. Haben Sie die Eingabe bereits früher durch Sternchen abgebrochen, druckt das Programm Leerzeilen.

Das können Sie vermeiden, indem Sie eine Zeile 677 mit der Anweisung LET X = I einfügen. Sie merken sich dann unter X die Zahl der eingetippten Zeilen.

Obere Schleifenbegrenzung gegen Leerzeilen

Dieses X verwenden Sie dann als obere Schleifenbegrenzung in 754. Sie ersetzen also die 30 durch X.

Ähnlich können Sie übrigens auch das Problem mit kürzeren Anschriften lösen, wenn Ihnen die vorhin beschriebene Möglichkeit zu umständlich erscheint. Natürlich müssen Sie dafür eine andere Variable verwenden.

Wenn das Programm in Zeile 772 angekommen ist und Sie fragt, ob Sie den Brief noch an weitere Anschriften schicken wollen, ist der Brief komplett.

Geben Sie in der INPUT-Anweisung Y$ den Wert "JA", können Sie eine weitere Anschrift eingeben und das Drucken beginnt von vorn. Bei allen von "JA" verschiedenen Eingaben verabschiedet sich das Programm.

Sollten Sie übrigens einmal versehentlich am Programmende landen, können Sie das Programm durch Eingabe von GOTO 612 ohne Zeilennummer davor fortsetzen und eine neue Anschrift eingeben.



```
100 REM ***********************************************************
110 REM *  Nußknacker Programm   Brief          (49)              *
120 REM *                                                         *
130 REM *  1. Teil   -   Initialisierung                          *
140 REM ***********************************************************
150 PRINT "HALLO, HIER IST DAS"
160 PRINT "NUSSKNACKER PROGRAMM"
170 PRINT "MIT DEM SIE IHRE"
180 PRINT "PERSOENLICHEN BRIEFE"
190 PRINT "SCHREIBEN KOENNEN"
200 PRINT
210 REM ***********************************************************
220 REM *  Platz für die Anschrift (A$)                           *
230 REM *  Platz für den Text (B$)                                *
240 REM ***********************************************************
250 DIM A$(6)
260 DIM B$(30)
270 REM ***********************************************************
280 REM *  2. Teil   -   Eingabe des Textes                       *
290 REM ***********************************************************
300 PRINT "SIE KOENNEN JETZT"
310 PRINT "DEN TEXT EINGEBEN"
320 PRINT
330 PRINT "ER KANN MAXIMAL"
340 PRINT "30 ZEILEN LANG SEIN"
350 PRINT
360 PRINT "IN JEDE ZEILE PASSEN"
370 PRINT "MAXIMAL 60 BUCHSTABEN"
380 PRINT
390 PRINT "WENN DER TEXT ZU ENDE IST"
400 PRINT "GEBEN SIE EINFACH * EIN"
410 PRINT
420 PRINT "LOS GEHT ES"
430 REM ***********************************************************
440 REM *  Hier beginnt die Schleife                              *
450 REM ***********************************************************
460 FOR I = 1 TO 30
470 PRINT
480 INPUT Z$
490 IF Z$="*" THEN GOTO 612
```

Fortsetzung auf der nächsten Seite ------------------------>

```
500 IF LEN(Z$) > 60 THEN GOTO 540
510 LET B$(I)=Z$
520 NEXT I
530 GOTO 612
540 PRINT
550 PRINT "ZEILE ZU LANG"
560 PRINT
570 PRINT "NEU EINGEBEN"
580 GOTO 470
602 REM *****************************************************
604 REM *  Der Text ist komplett                             *
606 REM *  3. Teil   -   Eingabe der Adresse                 *
608 REM *****************************************************
612 PRINT
614 PRINT "SIE KOENNEN JETZT"
616 PRINT "EINE ADRESSE EINGEBEN"
618 PRINT
622 PRINT "SIE KANN MAXIMAL"
624 PRINT "6 ZEILEN LANG SEIN"
626 PRINT
627 PRINT "IN JEDE ZEILE PASSEN"
632 PRINT "MAXIMAL 40 BUCHSTABEN"
634 PRINT
636 PRINT "WENN DIE ADRESSE ZU ENDE IST"
638 PRINT "GEBEN SIE EINFACH * EIN"
642 PRINT
644 PRINT "LOS GEHT ES"
646 REM *****************************************************
648 REM *  Hier beginnt die Schleife                         *
652 REM *****************************************************
664 FOR I = 1 TO 6
666 PRINT
668 INPUT Z$
672 IF Z$="*" THEN GOTO 706
674 IF LEN(Z$) > 40 THEN GOTO 684
676 LET A$(I)=Z$
678 NEXT I
682 GOTO 706
684 PRINT
686 PRINT "ZEILE ZU LANG"
```

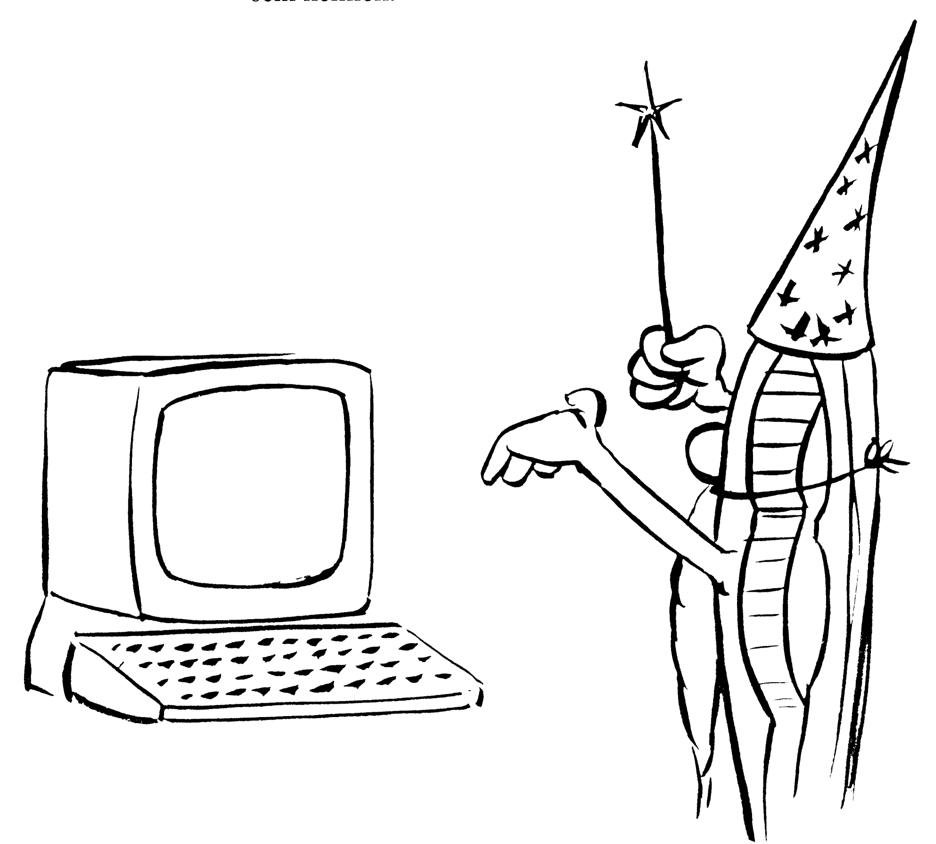
Fortsetzung auf der nächsten Seite ------------------------------->

```
688 PRINT
692 PRINT "NEU EINGEBEN"
694 GOTO 666
696 REM ***************************************************
698 REM *  Die Adresse ist komplett                        *
702 REM *  4. Teil   -   Drucken des Briefes               *
704 REM ***************************************************
706 OPEN 1,4
708 PRINT "BRIEF UND ADRESSE SIND KOMPLETT"
712 PRINT "JETZT WIRD GEDRUCKT"
714 PRINT
716 REM ***************************************************
718 REM *  zuerst die adresse                              *
722 REM ***************************************************
724 FOR I = 1 TO 6
726 PRINT#1,A$(I)
728 NEXT I
732 REM ***************************************************
734 REM *  dann die Leerzeilen                             *
736 REM ***************************************************
738 PRINT#1,
742 PRINT#1,
744 PRINT#1,
745 PRINT#1,
746 REM ***************************************************
748 REM *  und schließlich der Text                        *
752 REM ***************************************************
754 FOR I= 1 TO 30
756 PRINT#1,B$(I)
758 NEXT I
762 REM ***************************************************
764 REM *  Der Brief ist fertig gedruckt                   *
766 REM *  Neue Adresse ?                                  *
768 REM ***************************************************
769 CLOSE 1
772 PRINT "DER BRIEF IST FERTIG"
774 PRINT
776 PRINT ".SOLL ER NOCH AN EINE ADRESSE (JA/NEIN)?"
778 INPUT Y$
782 IF Y$="JA" THEN GOTO 612
784 PRINT
786 PRINT "DAS BRIEFPROGRAMM VERABSCHIEDET SICH"
788 REM ***************************************************
792 REM *  Programmende                                    *
794 REM ***************************************************
```

Geheimnisse gab es schon immer, und mit diesem Programm geben wir Ihnen eine Möglichkeit, Sie zu wahren. Sie können damit Texte verschlüsseln.

Machen Sie's geheimnisvoll

Abhängig ist die Kodierung von einer persönlichen Geheimzahl. Die müssen Sie beim Verschlüsseln eingeben und beim Entschlüsseln kennen.



Im Speicher müssen Sie Platz für den zu verschlüsselnden Text (T$) und die Länge der einzelnen Zeilen reservieren (A).

Zuerst geben Sie in der Schleife ab 460 den Text ein. Er wird unter Z$ zwischengespeichert. Wenn Sie keine 30 Zeilen eingeben wollen, können Sie die Eingabe des Textes mit * abbrechen. Der Stern wird natürlich nicht gespeichert.

Die Länge der Zeile ist wichtig

Unter X merken Sie sich die Anzahl der eingegebenen Zeilen. Das ist später für die Auswertung wichtig.

Das Programm akzeptiert maximal 60 Zeichen pro Zeile. Die Überprüfung erfolgt in 500. Ist die Zeile länger, wird Sie abgelehnt und erneute Eingabe angefordert (540—580).

Sind 30 Zeilen eingegeben oder die Eingabe mit * abgeschlossen, fordert das Programm ab Zeile 612 Ihre persönliche Geheimzahl an, die zwischen 2 und 999 liegen sollte.

Die Geheimzahl muß stimmen

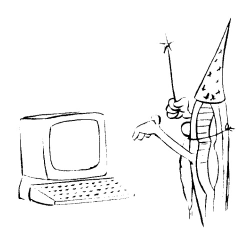
Das Programm prüft Ihre in 668 gemachte Eingabe auf Einhaltung dieses Bereiches. Wollen Sie einen anderen Bereich für Ihre Geheimzahl wählen, müssen Sie die Zahlen in der IF-Abfrage (Zeile 670) ändern.

Zahlen außerhalb des dort vorgegebenen Bereiches akzeptiert das Programm nicht. Die Geheimzahl wird unter G gespeichert.

Ab Zeile 724 beginnt dann die eigentliche Kodierung. Die Schleife läuft über alle Zeilen.

Wieviele Zeilen es sind, haben wir uns ja in X gemerkt.

Verschlüsseln der Zeilenlänge



Zuerst wird die Länge der Zeile durch Addition der Geheimzahl verschlüsselt und ausgegeben. Die Länge jeder Zeile steht ja im Feld A. Der Index (I) entspricht der Nummer der Zeile.

In der Schleife ab 738 findet die Verschlüsselung des eigentlichen Textes statt. Die Schleife ist eine Unterschleife zu der aus Zeile 724.

Ineinander geschachtelte Schleifen

Während die erste Schleife über alle Zeilen läuft, wird die zweite in jeder Zeile so oft durchlaufen, wie Buchstaben in der Zeile sind.

Dann wird die durch den Computer vorgegebene Kodierung der einzelnen Zeichen abgefragt und in R gespeichert. Dafür gibt es eine Standardfunktion (Zeile 744).

Zu dem so erhaltenen Wert addieren wir wieder die Geheimzahl und geben das Ergebnis aus.

Durch die Ineinanderschachtelung der beiden Schleifen haben wir später für jede Zeile die kodierte Angabe Ihrer Länge und den Kode jedes Zeichens.

```
100 REM ***********************************************************
110 REM *  Nußknacker Programm   Kodierung     (411)              *
120 REM *                                                         *
130 REM *  1. Teil   -   Begrüßung   -   Speicher                 *
140 REM ***********************************************************
150 PRINT "HALLO, HIER IST DAS"
160 PRINT "NUSSKNACKER PROGRAMM"
170 PRINT "MIT DEM SIE TEXTE"
180 PRINT "***   VERSCHLUESSELN   ***"
190 PRINT "KOENNEN"
200 PRINT
210 REM ***********************************************************
230 REM *  Platz für den Text (T$)                                *
235 REM *  Zeilenlänge (A)                                        *
240 REM ***********************************************************
250 DIM A(30)
260 DIM T$(30,60)
270 REM ***********************************************************
280 REM *  2. Teil   -   Eingabe des Textes                       *
290 REM ***********************************************************
300 PRINT "SIE KOENNEN JETZT"
310 PRINT "DEN TEXT EINGEBEN."
330 PRINT "ER KANN MAXIMAL"
340 PRINT "30 ZEILEN LANG SEIN."
360 PRINT "IN JEDE ZEILE PASSEN"
370 PRINT "MAXIMAL 60 BUCHSTABEN."
390 PRINT "WENN DER TEXT ZU ENDE IST"
400 PRINT "GEBEN SIE EINFACH * EIN."
410 PRINT
420 PRINT "LOS GEHT ES"
460 FOR I = 1 TO 30
465 LET X=I-1
470 PRINT
480 INPUT Z$
482 PRINT
484 PRINT Z$
490 IF Z$="*" THEN GOTO 612
500 IF LEN(Z$) > 60 THEN GOTO 540
510 LET T$(I)=Z$
512 LET A(I)=LEN(Z$)
520 NEXT I
525 X=X+1
530 GOTO 612
540 PRINT
```

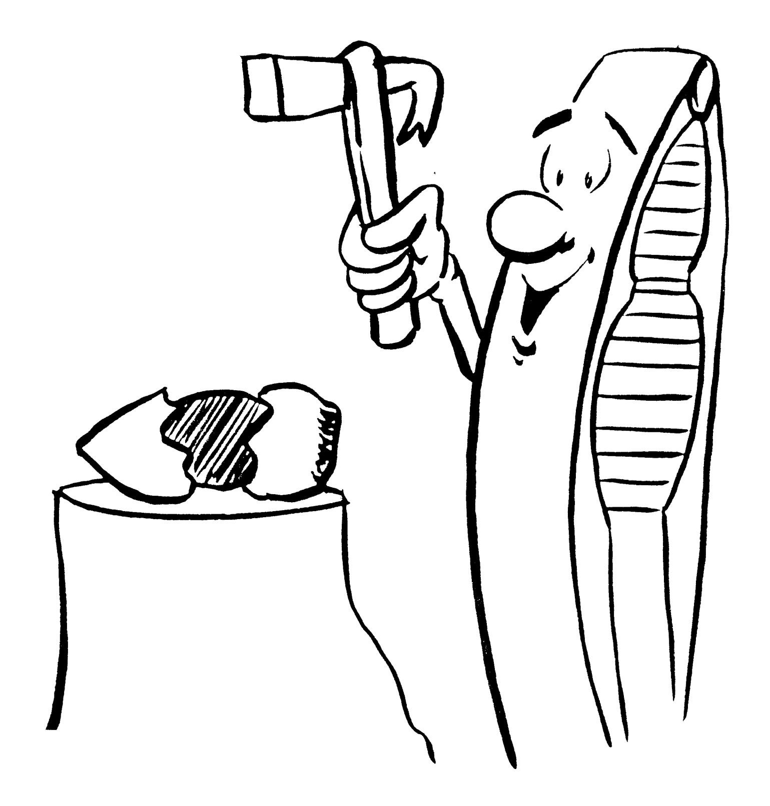
Fortsetzung auf der nächsten Seite ------------------------>

```
550 PRINT "ZEILE ZU LANG"
560 PRINT
570 PRINT "NEU EINGEBEN"
580 GOTO 470
602 REM ******************************************************
606 REM *  3. Teil   -   Geheimzahl                          *
608 REM ******************************************************
612 PRINT
614 PRINT "SIE KOENNEN JETZT EINE"
616 PRINT "GEHEIMZAHL EINGEBEN"
618 PRINT
622 PRINT "SIE DARF ZWISCHEN"
624 PRINT "2 UND 999 LIEGEN"
668 INPUT G
670 IF G<2 OR G>999 THEN GOTO 686
682 GOTO 706
686 PRINT "FALSCHE GEHEIMZAHL"
692 PRINT "NEU EINGEBEN"
694 GOTO 614
696 REM ******************************************************
702 REM *  4. Teil   -   Kodieren und ausgeben               *
704 REM ******************************************************
706 PRINT
708 PRINT "TEXT UND GEHEIMZAHL SIND KOMPLETT"
712 PRINT "JETZT WIRD VERSCHLUESSELT"
714 OPEN 1,4
716 REM ******************************************************
718 REM *  Schleife über alle Zeilen                         *
722 REM ******************************************************
724 FOR I = 1 TO X
726 PRINT#1,A(I)+G,
728 REM ******************************************************
730 REM *  Schleife über alle Zeichen einer Zeile            *
732 REM ******************************************************
738 FOR J = 1 TO A(I)
742 LET R$=T$(I,J)
744 LET R=ASC(R$)
745 PRINT#1,R+G,
746 NEXT J
748 NEXT I
750 CLOSE 1
762 REM ******************************************************
764 REM *  Der Text ist kodiert und gedruckt                 *
766 REM *  Programmende                                      *
768 REM ******************************************************
```

Mit diesem Programm können Sie die vorher verschlüsselten Texte wieder entziffern. Dazu müssen Sie natürlich die Geheimzahl kennen.

Knacken Sie das Geheimnis

Ohne die richtige Zahl läuft gar nichts. Sie bekommen bestenfalls Unsinn auf den Schirm, oder das Programm bricht mit einer Fehlermeldung ab.



Zuerst müssen Sie im Speicher wieder Platz für den Text (T$) und für die jeweiligen Zeilenlängen (A) reservieren.

Dann fragt das Programm nach der Geheimzahl, die unter G gespeichert wird, wenn sie im vorgegebenen Bereich liegt.

Wenn Sie übrigens im Kodierungsprogramm den Bereich geändert haben, sollten Sie das in diesem Programm nicht vergessen. Sonst haben Sie sich selber ausgetrickst.

Hoffentlich haben Sie jetzt die richtige Geheimzahl eingegeben, denn ab Zeile 717 wird jetzt entschlüsselt.

Wieviel Zeilen hat der Text?

I ist der Index für die Schleife über alle Zeilen. Da wir noch nicht wissen, wieviel Zeilen der Text hat, können wir die Schleife diesmal nicht mit einer FOR-Schleife realisieren. I gibt jeweils an, in der wievielten Zeile wir uns befinden und wird im Programm bei 717 auf 0 gesetzt.

Nun können Sie den kodierten Text Zahl für Zahl eingeben. Als letzte Zahl müssen Sie zur Markierung des Textendes 9999 eingeben.

Wählen Sie 9999 zum Abschluß

Wenn Sie alle Zahlen richtig eingegeben haben, kann die 9999 nur an der Stelle kommen, an der das Programm eigentlich die Eingabe der kodierten Zeilenlänge erwartet.

Sie erinnern sich, wir hatten im Kodierungsprogramm die Zeilenlänge stets mitkodiert. So wissen wir jetzt immer, wann eine

Zeile zu Ende ist. Natürlich hätten wir auch einen Kode für Zeilenende festlegen können. Doch damit wäre der Kode leichter zu knakken.

Aus der ersten Zahl jeder Zeile können wir also ihre Länge ablesen, indem wir die Geheimzahl subtrahieren. Das passiert in Zeile 730.

Soviel Zeichen hat die Zeile

Damit haben wir gleich auch die Obergrenze für die Schleife, in der die einzelnen Zeichen einer Schleife dekodiert werden. Diese Schleife läuft von Zeile 738 bis 746.

Jede Zahl wird eingelesen, dann subtrahiert das Programm die Geheimzahl und wandelt das Ergebnis mit der Standardfunktion CHR$ in ein Textzeichen um, das wir in Feld T$ (für Text) speichern.

Ab Zeile 800 müssen wir dann den fertig dekodierten Text nur noch ausgeben. Nach 800 kommt das Programm, wenn Sie 9999 eingeben.

Die Zahl der Zeilen haben wir uns ja in I gemerkt. Weil aber die 9999-Zeile mitgezählt wurde, müssen wir 1 abziehen, um die Obergrenze für die Schleife ab 800 zu erhalten.

Zum Schluß druckt das Programm noch die Anzahl der dekodierten Zeilen. Versuchen Sie doch mal, ein Programm zu schreiben, das den Kode ohne Geheimzahl knackt.



```
100 REM ***********************************************************
110 REM *  Nußknacker Programm   Dekodierung    (412)            *
120 REM *                                                         *
130 REM *  1. Teil   -   Begrüßung   -   Speicher                 *
140 REM ***********************************************************
150 PRINT "HALLO, HIER IST DAS"
160 PRINT "NUSSKNACKER PROGRAMM"
170 PRINT "MIT DEM SIE TEXTE"
175 PRINT
180 PRINT "***   ENTSCHLUESSELN   ***"
185 PRINT
190 PRINT "KOENNEN"
200 PRINT
210 REM ***********************************************************
230 REM *  Platz für den Text (T$)                                *
235 REM *  Zeilenlänge (A)                                        *
240 REM ***********************************************************
250 DIM A(30)
260 DIM T$(30,60)
602 REM ***********************************************************
606 REM *  2. Teil   -   Geheimzahl                               *
608 REM ***********************************************************
610 PRINT
612 PRINT
614 PRINT "SIE KOENNEN JETZT EINE"
616 PRINT "GEHEIMZAHL EINGEBEN"
618 PRINT
622 PRINT "SIE DARF ZWISCHEN"
624 PRINT "2 UND 999 LIEGEN"
668 INPUT G
670 IF G<2 OR G>999 THEN GOTO 686
682 GOTO 706
686 PRINT "FALSCHE GEHEIMZAHL"
692 PRINT "NEU EINGEBEN"
693 PRINT
694 GOTO 618
696 REM ***********************************************************
702 REM *  3. Teil   -   Dekodieren und speichern                 *
704 REM ***********************************************************
706 PRINT
707 PRINT "JETZT WIRD ENTSCHLUESSELT"
```

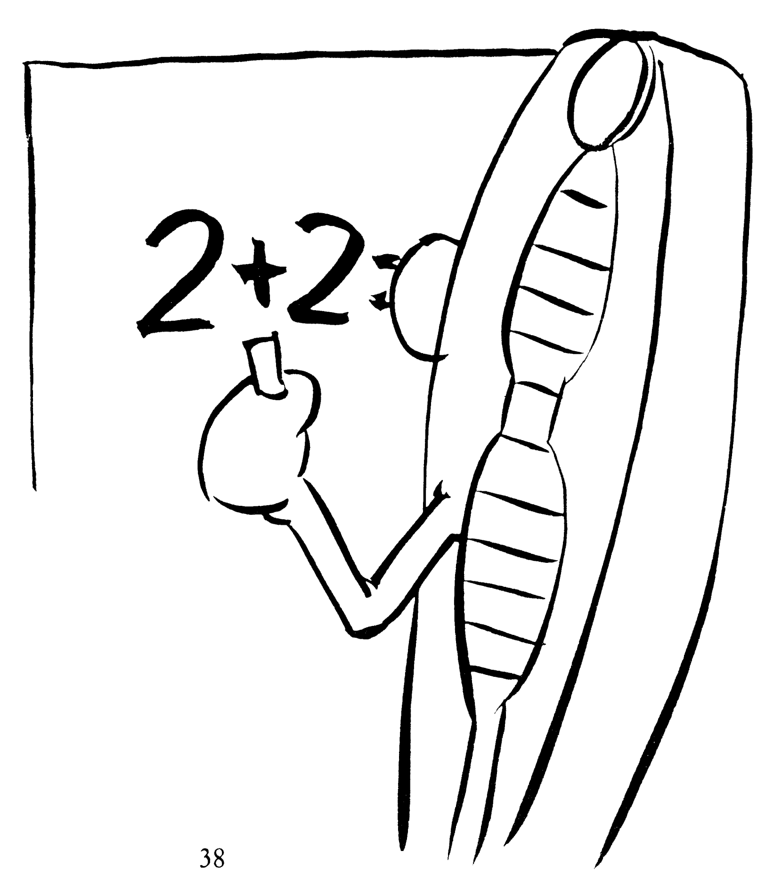
Fortsetzung auf der nächsten Seite ------------------------->

```
708 PRINT
709 PRINT "GEBEN SIE ALS LETZTE"
710 PRINT "ZAHL 9999 an"
711 PRINT
712 PRINT "DER ORIGINALTEXT DARF"
713 PRINT "MAXIMAL 30 ZEILEN HABEN"
714 PRINT
715 PRINT "DER ORIGINALTEXT DARF PRO ZEILE"
716 PRINT "MAXIMAL 60 ZEICHEN HABEN"
717 LET I=0
718 REM **********************************************************
719 REM *  Schleife über alle Zeilen                              *
722 REM **********************************************************
724 LET I=I+1
726 INPUT A(I)
728 IF A(I)=9999 THEN GOTO 800
730 LET A(I)=A(I)-G
732 REM **********************************************************
734 REM *  Schleife über alle Zeichen einer Zeile                 *
736 REM **********************************************************
738 FOR J = 1 TO A(I)
740 INPUT R
742 LET R=R-G
744 LET R$=CHR$(R)
745 LET T$(I,J)=R$
746 NEXT J
748 GOTO 724
762 REM **********************************************************
764 REM *  Der Text ist dekodiert und gespeichert                 *
766 REM *  4. Teil   -   Ausdrucken des Textes                    *
768 REM **********************************************************
800 FOR L=1 TO I-1
810 LPRINT T$(L)
820 NEXT L
822 LPRINT
824 LPRINT
826 LPRINT I-1;" DEKODIERTE TEXTZEILEN"
830 REM **********************************************************
840 REM *  Programmende                                           *
850 REM **********************************************************
```

Dieses Programm trainiert Sie in den Grundrechenarten. Sie müssen mindestens so lange trainieren, bis Sie 30 Aufgaben aus jedem Bereich hintereinander fehlerfrei lösen.

Computer rechnen meist richtig

Und denken Sie daran. Computer irren sich beim Rechnen selten, wenn sie richtig programmiert sind. Also geben Sie nicht dem Computer die Schuld.



Zuerst erhalten im Programm nach der Begrüßung die Statistik-Variablen den Grundwert 0.

Unter A1 zählt der Computer später die Additionsaufgaben, die er Ihnen gestellt hat. Unter A2 die richtigen Antworten. Beide sollten nach Möglichkeit immer den gleichen Wert haben, dann sind Sie richtig gut.

Unter S1 zählt der Computer später die Subtraktionsaufgaben, die er Ihnen gestellt hat. Unter S2 die richtigen Antworten. Auch hier sollten beide Variable nach Möglichkeit immer den gleichen Wert haben.

Grundwerte für die Statistik

Unter M1 schließlich zählt der Computer später die Multiplikationsaufgaben, die er Ihnen gestellt hat. Unter M2 die richtigen Antworten. Wenn Sie es schaffen, auch diese beiden Werte immer in Übereinstimmung zu halten, sind Sie Spitze.

Übrigens, wenn Sie einen Taschenrechner neben Ihren Computer legen, macht das Spiel nur halb so viel Spaß. Trainieren Sie lieber Ihre grauen Zellen als die Speicherzellen des Taschenrechners.

Wählen Sie die Rechenart

Als nächstes können Sie im Programm die gewünschte Rechenart auswählen. Geben Sie für die Addition ein "A", für die Subtraktion ein "S" und für die Multiplikation ein "M" ein. Ab Zeile 380 prüft der Computer Ihre Eingabe und das Programm wird entsprechend

fortgesetzt. Wenn Sie ein anderes Zeichen eingeben, als eins der drei erlaubten, weist der Computer Sie „sanft“ aber konsequent auf den richtigen Weg.

Trainieren Sie die Addition

In die Zeile 500 kommen Sie, wenn Sie das Additionstraining gewählt haben. Zuerst merkt der Computer sich, die wievielte Aufgabe er Ihnen bei der Addition stellt. Dann werden in den Zeilen 516 bis 526 zwei Zufallszahlen bestimmt.

Zufallszahlen liegen immer zwischen 0 und 1. Solche Werte, mit vielen Stellen hinter dem Komma, sind für das Training nicht geeignet.

Durch Multiplikation mit 1000 bringen wir die erste Zahl in den Bereich von 0 bis 999. Dann schneiden wir die Reste hinter dem Komma ab (Funktion INT).

Die zweite Zahl wählen wir nicht ganz so hoch und legen sie durch Multiplikation mit 100 in den Bereich von 0 bis 99.

Die Schwierigkeit bestimmen Sie

Den Schwierigkeitsgrad können Sie übrigens durch Veränderung dieser Werte selbst bestimmen.

Endlich ist es soweit, und der Computer nennt Ihnen die beiden Werte, die Sie addieren sollen.

Gleich vergleicht er Ihre Eingabe mit dem richtigen Ergebnis (Zeile 540). Wenn Sie richtig liegen, addiert er bei den richtigen Lösungen (A2) einen Treffer.

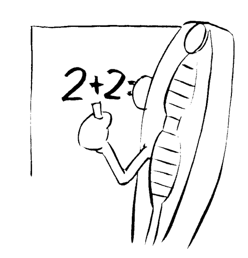
Sonst gibt er Ihnen zum Trost das richtige Ergebnis. Aus Fehlern wird man klug.

Sie können sich jetzt nach einem Sprung zu Zeile 280 für einen neuen oder denselben Trainingsbereich entscheiden und sich eine neue Aufgabe stellen lassen.

Oder Sie haben die Schnauze voll und geben bei 568 "NEIN" ein, dann liefert das Programm Ihnen ab Zeile 800 eine Statistik Ihrer Leistungen.

Subtrahieren will gelernt sein

Der nächste Programmteil ist die Subtraktion. In die Zeile 600 kommen Sie, wenn Sie das Subtraktionstraining gewählt haben.



Zuerst merkt der Computer sich, die wievielte Aufgabe er Ihnen bei der Subtraktion stellt. Dann werden in den Zeilen 616 bis 626 zwei Zufallszahlen bestimmt. Das Verfahren entspricht dem bei der Addition.

Auch hier können Sie den Schwierigkeitsgrad durch Veränderung der Werte 1000 und 100 selbst bestimmen.

Zahlenpaare auf dem Bildschirm

Endlich ist es dann soweit, und der Computer nennt Ihnen die beiden Werte. Sie sollen den zweiten vom ersten abziehen.

Gleich vergleicht er Ihre Eingabe mit dem richtigen Ergebnis (Zeile 640). Wenn Sie richtig liegen, addiert er bei den richtigen Lösungen (diesmal S2) einen Treffer. Sonst gibt er Ihnen zum Trost das richtige Ergebnis.

Sie können sich jetzt nach einem Sprung zu Zeile 280 für einen neuen oder denselben Trainingsbereich entscheiden und sich eine neue Aufgabe stellen lassen.

Wenn Sie keine Lust mehr haben, bringt ein ”NEIN” in Zeile 668 Sie zur Statistik in Zeile 800.

Multiplizieren ist ganz schön schwer

Der letzte Trainingsteil ist die Multiplikation. Er beginnt ab Zeile 700. Dahin kommen Sie, wenn sie ein ”M” für Multiplikation eingegeben haben.

Zuerst merkt der Computer sich, die wievielte Aufgabe er Ihnen bei der Multiplikation stellt. Dann werden in den Zeilen 716 bis 726 zwei Zufallszahlen bestimmt.

Wieder können Sie den Schwierigkeitsgrad selbst bestimmen. Der Computer nennt Ihnen dann die beiden Werte, die Sie multiplizieren sollen.

Haben Sie einen Treffer zu verzeichnen?

Gleich vergleicht er Ihre Eingabe mit dem richtigen Ergebnis (Zeile 740). Wenn Sie richtig liegen, addiert er bei den richtigen Lösungen (M2) einen Treffer. Sonst bekommen Sie das richtige Ergebnis.

Sie können dann das Training abbrechen und sich die statistischen Werte Ihrer Rechenkünste nennen lassen, oder Sie lassen den Computer eine neue Aufgabe stellen.

Durch die Zufallsfunktion des Systems

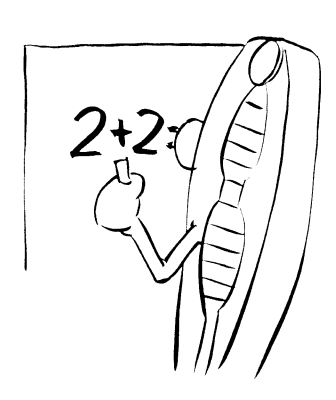
werden Ihre Trainingsstunden übrigens nie gleich verlaufen. In Zeile 260 wird die Basis für Ihre Trainingszahlen abhängig von der Zeit, die Ihr Computer schon eingeschaltet ist, gelegt.

Verfeinern Sie die Bewertung

Der Statistikteil am Ende des Programms (ab Zeile 800) liefert Ihnen dann die Einzelergebnisse in den „Disziplinen". Zusätzlich addiert er die Einzelergebnisse.

Dieses Gesamtergebnis erhält ab Zeile 854 eine Wertung. Da ist noch viel Raum für Ihre Phantasie, denn die Wertung kann durchaus noch wesentlich differenzierter ausfallen.

Zum Abschluß können Sie wählen, ob Sie noch einmal ganz von vorn beginnen wollen. Alle statistischen Werte werden dann zurückgesetzt. Die Alternative ist das Ende des Programms.



```
100 REM ***********************************************************
110 REM *  Nußknacker Programm   Training        (46)             *
120 REM *                                                         *
130 REM *  1. Teil   -   Basisdaten   -   Initialisierung         *
140 REM ***********************************************************
150 PRINT "HALLO, HIER IST DAS"
160 PRINT "NUSSKNACKER PROGRAMM"
170 PRINT "FUER IHR PERSOENLICHES"
180 PRINT
190 PRINT "*****   TRAINING  *****"
200 LET A1=0
210 LET A2=0
220 LET S1=0
230 LET S2=0
240 LET M1=0
250 LET M2=0
260 LET X=RND(-TI)
270 PRINT
280 PRINT "WAS WOLLEN SIE TRAINIEREN ?"
290 PRINT "ZUR AUSWAHL HABEN SIE:"
300 PRINT
310 PRINT "ADDIEREN        (A)"
320 PRINT "SUBTRAHIEREN    (S)"
330 PRINT "MULTIPLIZIEREN  (M)"
340 INPUT A$
350 REM ***********************************************************
360 REM *  2. Teil   -   Auswahl                                  *
370 REM ***********************************************************
380 IF A$="A" THEN GOTO 500
390 IF A$="S" THEN GOTO 600
400 IF A$="M" THEN GOTO 700
410 PRINT
420 PRINT "DIESE EINGABE WAR NICHT RICHTIG"
430 PRINT "PROBIEREN SIE ES NOCHMAL"
440 PRINT
450 GOTO 290
500 REM ***********************************************************
502 REM *  3. Teil   -   Addieren                                 *
503 REM ***********************************************************
504 PRINT
506 PRINT "TRAINIEREN SIE DIE ADDITION"
```

Fortsetzung auf der nächsten Seite ------------------------>

```
508 LET A1=A1+1
510 PRINT "AUFGABE NR. ";A1
512 PRINT
514 PRINT "ADDIEREN SIE DIESE ZAHLEN:"
516 LET X=RND(0)
518 LET X=X*1000
520 LET X=INT(X)
522 LET Y=RND(0)
524 LET Y=Y*100
526 LET Y=INT(Y)
528 PRINT X,Y
530 PRINT
532 PRINT "SAGEN SIE MIR DAS ERGEBNIS"
534 INPUT E
536 PRINT E
538 PRINT
540 IF E=X+Y THEN GOTO 550
542 PRINT "LEIDER FALSCH"
544 PRINT "DAS RICHTIGE ERGEBNIS IST:"
546 PRINT X+Y
548 GOTO 556
550 PRINT "TREFFER"
552 PRINT "SIE HABEN DIE RICHTIGE LOESUNG"
554 LET A2=A2+1
556 PRINT
558 PRINT "STATISTIK ADDITION:"
560 PRINT "GESTELLTE AUFGABEN: ";A1
562 PRINT "RICHTIG GELOESTE AUFGABEN: ";A2
564 PRINT
566 PRINT "WOLLEN SIE WEITERMACHEN (JA/NEIN)?"
568 INPUT A$
570 IF A$="NEIN" THEN GOTO 800
571 PRINT
572 GOTO 280
600 REM *********************************************************
602 REM *  4. Teil   -   Subtrahieren                            *
603 REM *********************************************************
604 PRINT
606 PRINT "TRAINIEREN SIE DIE SUBTRAKTION"
608 LET S1=S1+1
610 PRINT "AUFGABE NR. ";S1
```

Fortsetzung auf der nächsten Seite ------------------------>

```
612 PRINT
614 PRINT "SUBTRAHIEREN SIE ZAHL 2 VON ZAHL 1"
616 LET X=RND(0)
618 LET X=X*1000
620 LET X=INT(X)
622 LET Y=RND(0)
624 LET Y=Y*100
626 LET Y=INT(Y)
628 PRINT X,Y
630 PRINT
632 PRINT "SAGEN SIE MIR DAS ERGEBNIS"
634 INPUT E
636 PRINT E
638 PRINT
640 IF E=X-Y THEN GOTO 650
642 PRINT "LEIDER FALSCH"
644 PRINT "DAS RICHTIGE ERGEBNIS IST:"
646 PRINT X-Y
648 GOTO 656
650 PRINT "TREFFER"
652 PRINT "SIE HABEN DIE RICHTIGE LOESUNG"
654 LET S2=S2+1
656 PRINT
658 PRINT "STATISTIK SUBTRAKTION"
660 PRINT "GESTELLTE AUFGABEN: ";S1
662 PRINT "RICHTIG GELOESTE AUFGABEN: ";S2
664 PRINT
666 PRINT "WOLLEN SIE WEITERMACHEN (JA/NEIN)?"
668 INPUT A$
670 IF A$="NEIN" THEN GOTO 800
671 PRINT
672 GOTO 280
700 REM ***********************************************************
702 REM *  5. Teil   -   Multiplizieren                            *
703 REM ***********************************************************
704 PRINT
706 PRINT "TRAINIEREN SIE DIE MULTIPLIKATION"
708 LET M1=M1+1
710 PRINT "AUFGABE NR. ";M1
712 PRINT
714 PRINT "MULTIPLIZIEREN SIE DIESE ZAHLEN:"
```

Fortsetzung auf der nächsten Seite ------------------------->

```
716 LET X=RND(0)
718 LET X=X*1000
720 LET X=INT(X)
722 LET Y=RND(0)
724 LET Y=Y*100
726 LET Y=INT(Y)
728 PRINT X,Y
730 PRINT
732 PRINT "SAGEN SIE MIR DAS ERGEBNIS"
734 INPUT E
736 PRINT E
738 PRINT
740 IF E=X*Y THEN GOTO 750
742 PRINT "LEIDER FALSCH"
744 PRINT "DAS RICHTIGE ERGEBNIS IST:"
746 PRINT X*Y
748 GOTO 756
750 PRINT "TREFFER"
752 PRINT "SIE HABEN DIE RICHTIGE LOESUNG"
754 LET M2=M2+1
756 PRINT
758 PRINT "STATISTIK MULTIPLIKATION:"
760 PRINT "GESTELLTE AUFGABEN: ";M1
762 PRINT "RICHTIG GELOESTE AUFGABEN: ";M2
764 PRINT
766 PRINT "WOLLEN SIE WEITERMACHEN (JA/NEIN)?"
768 INPUT A$
770 IF A$="NEIN" THEN GOTO 800
771 PRINT
772 GOTO 280
800 REM ***********************************************************
802 REM *  6. Teil   -   Gesamtstatistik                          *
804 REM ***********************************************************
806 PRINT
808 PRINT "GESAMTSTATISTIK"
810 PRINT
812 PRINT "ADDITION:"
814 PRINT "GESTELLTE AUFGABEN: ";A1
816 PRINT "RICHTIGE LOESUNGEN: ";A2
818 PRINT
820 PRINT "SUBTRAKTION:"
```

Fortsetzung auf der nächsten Seite ------------------------>

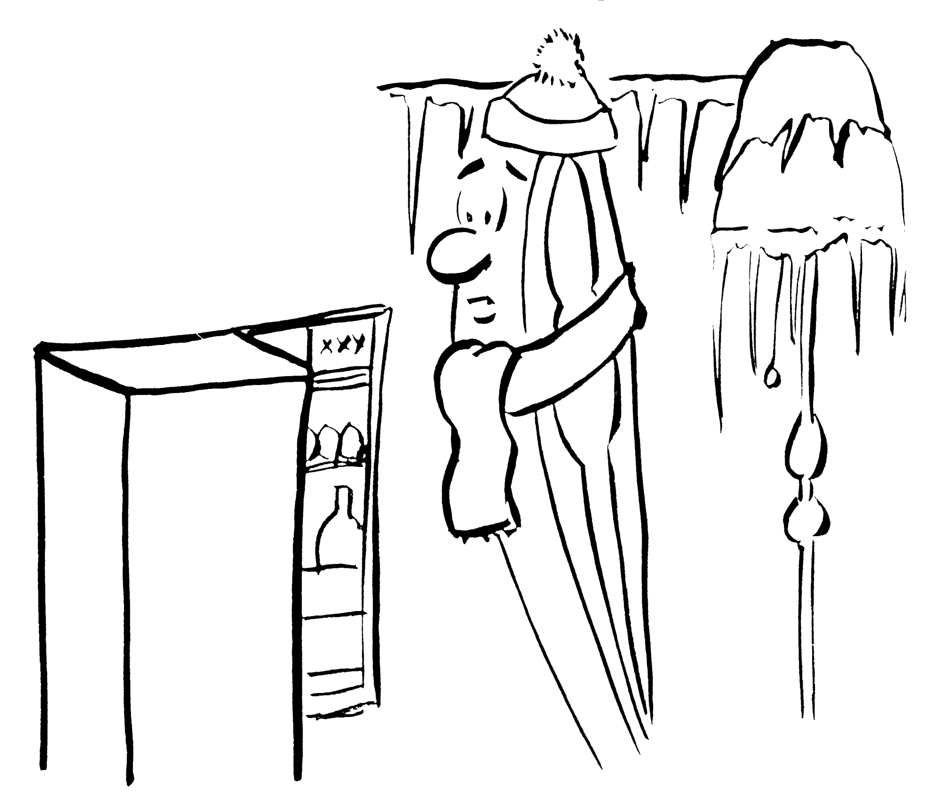
```
822 PRINT "GESTELLTE AUFGABEN: ";S1
824 PRINT "RICHTIGE LOESUNGEN: ";S2
826 PRINT
828 PRINT "MULTIPLIKATION:"
830 PRINT "GESTELLTE AUFGABEN: ";M1
832 PRINT "RICHTIGE LOESUNGEN: ";M2
834 PRINT
836 LET G1=A1+S1+M1
838 LET G2=A2+S2+M2
840 PRINT "INSGESAMT:"
842 PRINT "GESTELLTE AUFGABEN: ";G1
844 PRINT "RICHTIGE LOESUNGEN: ";G2
846 PRINT
848 REM *****************************************************
850 REM *  7. Teil   -   Bewertung                            *
852 REM *****************************************************
854 IF G1=G2 THEN PRINT "GROSSARTIG"
856 IF G1-G2 < 5 THEN PRINT "MITTELMAESSIG"
858 IF G1-G2 > 4 THEN PRINT "ZIEMLICH MIES"
860 PRINT
862 PRINT "WOLLEN SIE EINEN NEUEN VERSUCH STARTEN (JA/NEIN)?"
864 INPUT A$
866 IF A$="JA" THEN GOTO 100
868 PRINT
870 PRINT "DANN EBEN NICHT"
872 REM *****************************************************
874 REM *  Programmende                                       *
876 REM *****************************************************
```

```
764*934   8236+590   8876-654   764*934   8236+590   8876-654
 764*934   8236+590   8876-654   764*934   8236+590   8876-654
  764*934   8236+590   8876-654   764*934   8236+590   8876-654
64*934   8236+590   8876-654   764*934   8236+590   8876-654
 764*934   8236+590   8876-654   764*934   8236+590   8876-654
64*934   8236+590   8876-654   764*934   8236+590   8876-654
  764*934   8236+590   8876-654   764*934   8236+590   8876-654
```

Die Zuordnung eines Landes zu einer vorgegebenen Stadt können Sie mit diesem Programm üben. Wir haben im Beispiel nur wenige Städte und Länder vorgegeben. Damit haben Sie aber erst einmal einen Anfang.

Wo ist es bitter kalt?

Später können Sie dann ergänzen. Eine Erweiterung dieses Programms mit Lösungsvorschlägen finden Sie im nächsten Kapitel.



Zuerst müssen Sie im Speicher Platz für Ihre Daten schaffen. Da sind einmal die Daten, nach denen Sie fragen wollen (S$). Dazu gehören die Lösungen (L$).

Wir haben in unserem Beispielprogramm sechs solcher Paare vorgesehen. Entsprechend sehen die DIM-Anweisungen in den Zeilen 230 und 240 Platz für sechs Worte vor.

Wenn Sie dieses Programm ergänzen wollen, um mehr Fragen unterzubringen, müssen Sie statt der 6 die von Ihnen vorgesehene Anzahl eintragen.

Lösung und Suchwort haben gleichen Index

In den folgenden Anweisungen (bis 370) werden zuerst sämtliche Suchworte eingegeben, dann sämtliche Lösungen. Die Zahl in Klammern (Index) muß für Suchwort und Lösung immer gleich sein. Es unterscheiden sich nur die Speicherbereiche (S$ oder L$).

Ab Zeile 420 beginnt dann die Abfrageschleife. Es werden alle Suchworte nacheinander abgefragt. Falls Sie die Anzahl der Suchworte geändert haben, müssen Sie in Zeile 420 die 6 ebenfalls ändern.

Keine Antwort wird vergessen

Ihre Antworten merkt sich der Computer übrigens auch. Dafür steht das Feld A$ (wie Antworten) zur Verfügung. Es muß genauso groß sein, wie die beiden anderen Felder. Die Größe wird in Zeile 250 festgelegt.

Die Antworten liest der Computer in Zeile 490. Er prüft, ob Sie die richtige Lösung ange-

geben haben. Falls nein, schreibt er die richtige Lösung auf den Bildschirm.

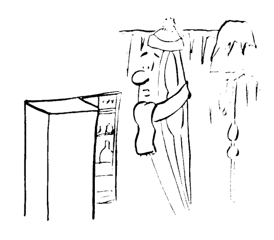
In Zeile 710 geht dann nach einer Abfrage, ob Sie überhaupt weiter raten wollen, die Schleife von Zeile 420 zu Ende. Ein Durchlauf entspricht dem Raten (oder Nicht-Raten) eines Suchwortes.

Alle Fragen, alle Antworten

Zum Schluß liefert der Computer Ihnen eine Gesamtliste aller Suchworte, Lösungen und Ihrer Antworten. Sie können dann noch einmal im Überblick Ihre Antworten mit den richtigen Lösungen vergleichen. In Zeile 840 müssen Sie wieder die 6 ändern, falls Sie zusätzliche Suchworte programmieren.

Trainieren Sie umgekehrt

Wenn Sie auch umgekehrt trainieren wollen, müssen Sie einfach überall im Programm nach Zeile 380 S$ und L$ vertauschen. Alternativ können Sie S$ und L$ auch in den Zeilen 260 bis 370 austauschen. Die Texte sollten Sie dann allerdings auch ändern, sonst sind die Fragen ziemlich unsinnig. Viel Spaß!



```
110 REM *******************************************************
120 REM *  Nußknacker Programm   Raten Sie mal  (43)          *
130 REM *                                                     *
140 REM *  1. Teil   -   Basisdaten   -   Initialisierung     *
150 REM *******************************************************
160 PRINT "HALLO, HIER IST DAS"
170 PRINT "NUSSKNACKER PROGRAMM"
180 PRINT
190 PRINT "RATEN SIE MAL"
200 PRINT
210 PRINT "ICH NENNE IHNEN JETZT STAEDTE"
220 PRINT "NENNEN SIE DAS LAND DAZU"
230 DIM S$(6)
240 DIM L$(6)
250 DIM A$(6)
260 LET S$(1)="PARIS"
270 LET S$(2)="LONDON"
280 LET S$(3)="NEW YORK"
290 LET S$(4)="LISSABON"
300 LET S$(5)="MADRID"
310 LET S$(6)="TOKIO"
320 LET L$(1)="FRANKREICH"
330 LET L$(2)="ENGLAND"
340 LET L$(3)="USA"
350 LET L$(4)="PORTUGAL"
360 LET L$(5)="SPANIEN"
370 LET L$(6)="JAPAN"
380 PRINT
390 REM *******************************************************
400 REM *  2. Teil   -   Abfrage                              *
410 REM *******************************************************
420 FOR I = 1 TO 6
430 PRINT440 PRINT "NUN RATEN SIE MAL"
450 PRINT
460 PRINT S$(I)
470 PRINT
480 PRINT "WIE HEISST DAS LAND DAZU?"
490 INPUT A$(I)
500 REM *******************************************************
510 REM *  3. Teil   -   Auswertung                           *
520 REM *******************************************************
530 IF A$(I)=L$(I) THEN GOTO 620
```

Fortsetzung auf der nächsten Seite ------------------------>

```
540 PRINT "DAS STIMMT LEIDER NICHT"
550 PRINT
560 PRINT "DIE RICHTIGE LOESUNG IST: ";
570 PRINT L$(I)
580 PRINT
590 PRINT "VERSUCHEN SIE ES NOCHMAL"
600 PRINT
610 GOTO 680
620 PRINT "TREFFER"
630 PRINT "GENAU DAS IST DIE LOESUNG"
640 PRINT
650 REM *********************************************************
660 REM *  4. Teil   -   Abfrage nochmal raten?                  *
670 REM *********************************************************
680 PRINT "WOLLEN SIE NOCHMAL RATEN (JA/NEIN)?"
690 INPUT Q$
700 IF Q$="NEIN" THEN GOTO 770
710 NEXT I
720 PRINT "LEIDER NICHTS MEHR DA"
730 PRINT
740 REM *********************************************************
750 REM *  5. Teil   -   Gesamtliste                             *
760 REM *********************************************************
770 PRINT "WOLLEN SIE EINE LISTE"
780 PRINT "IHRER RATEVERSUCHE (JA/NEIN)?"
790 INPUT Q$
800 IF Q$="NEIN" THEN GOTO 900
810 PRINT
820 PRINT "SO HABEN SIE GERATEN"
830 PRINT
840 FOR J = 1 TO 6
850 PRINT S$(J);" - ";
860 PRINT A$(J);" (";
870 PRINT L$(J);")"
880 NEXT J
890 PRINT
900 PRINT "NUN IST SCHLUSS"
910 REM *********************************************************
920 REM *  Programmende                                          *
930 REM *********************************************************
```

In diesem Ratespiel müssen Sie zu jedem Autotyp den Hersteller raten. Wir haben hier nur ein paar Autos vorgegeben. Damit können Sie erst mal loslegen.

Autos sind leicht geknackt

Später können Sie dann weitere Autos mit den Herstellern ergänzen. Das Programm schlägt Ihnen übrigens Lösungen vor.



Zuerst müssen Sie im Speicher Platz für Ihre Daten schaffen. Da sind einmal die Daten, nach denen Sie fragen wollen (S$). Dazu gehören die Lösungen (L$).

Wir haben in unserem Beispielprogramm sechs solcher Paare vorgesehen. Entsprechend sehen die DIM-Anweisungen in den Zeilen 230 und 240 Platz für sechs Worte vor.

Wenn Sie dieses Programm ergänzen wollen, um mehr Fragen unterzubringen, müssen Sie statt der 6 die von Ihnen vorgesehene Anzahl eintragen.

Erst das Suchwort, dann die Lösung

In den folgenden Anweisungen (bis 370) werden zuerst sämtliche Suchworte eingegeben, dann sämtliche Lösungen. Die Zahl in Klammern (Index) muß für Suchwort und Lösung immer gleich sein. Es unterscheiden sich nur die Speicherbereiche (S$ oder L$).

Ab Zeile 420 beginnt dann die Abfrageschleife. Es werden alle Suchworte nacheinander abgefragt. Falls Sie die Anzahl der Suchworte geändert haben, müssen Sie in Zeile 420 die 6 ebenfalls ändern.

Aus diesen Lösungen können Sie wählen

Damit das Raten nicht ganz so schwer ist, gibt das Programm in der Schleife ab Zeile 483 sämtliche möglichen Lösungen an. Die Zeile 484 ändert aber die Reihenfolge, damit man nicht sofort sieht, daß Lösung 5 beispielsweise zu Frage 5 gehört.

Diese Schleife wird bei jeder Frage neu

durchlaufen. Sie bekommen also zu jeder Frage alle möglichen Lösungen genannt.

Wenn Sie das Programm beispielsweise zum Vokabeltraining benutzen, hat das den zusätzlichen Vorteil, daß sich die Worte durch häufige Präsentation optisch einprägen.

Antworten im Speicher

Ihre Antworten merkt sich der Computer übrigens auch. Dafür steht das Feld A$ (wie Antworten) zur Verfügung. Es muß genauso groß sein, wie die beiden anderen Felder. Die Größe wird in Zeile 250 festgelegt.



Die Antworten liest der Computer in Zeile 490. Er prüft, ob Sie die richtige Lösung angegeben haben. Falls nein, schreibt er die richtige Lösung auf den Bildschirm.

In Zeile 710 geht dann nach einer Abfrage, ob Sie überhaupt weiter raten wollen, die Schleife von Zeile 420 zu Ende. Ein Durchlauf entspricht dem Raten (oder Nicht-Raten) eines Suchwortes.

Fragen und Antworten im Überblick

Zum Schluß liefert der Computer Ihnen eine Gesamtliste aller Suchworte, Lösungen und Ihrer Antworten. Sie können dann noch einmal im Überblick Ihre Antworten mit den richtigen Lösungen vergleichen. In Zeile 840 müssen Sie wieder die 6 ändern, falls Sie zusätzliche Suchworte programmieren.

```
110 REM ******************************************************
120 REM *  Nußknacker Programm   Autos knacken  (48)         *
130 REM *                                                    *
140 REM *  1. Teil   -   Basisdaten   -   Initialisierung    *
150 REM ******************************************************
160 PRINT "HALLO, HIER IST DAS"
170 PRINT "NUSSKNACKER PROGRAMM"
180 PRINT
190 PRINT "AUTOS KNACKEN"
200 PRINT
210 PRINT "ICH NENNE IHNEN JETZT AUTOTYPEN"
220 PRINT "NENNEN SIE DEN HERSTELLER DAZU"
230 DIM S$(6)
240 DIM L$(6)
250 DIM A$(6)
260 LET S$(1)="ESCORT"
270 LET S$(2)="GOLF"
280 LET S$(3)="MANTA"
290 LET S$(4)="RITMO"
300 LET S$(5)="207"
310 LET S$(6)="504"
320 LET L$(1)="FORD"
330 LET L$(2)="VW"
340 LET L$(3)="OPEL"
350 LET L$(4)="FIAT"
360 LET L$(5)="MERCEDES"
370 LET L$(6)="PEUGEOT"
380 PRINT
390 REM ******************************************************
400 REM *  2. Teil   -   Abfrage mit Vorschlägen             *
410 REM ******************************************************
420 FOR I = 1 TO 6
430 PRINT
440 PRINT "NUN RATEN SIE MAL"
450 PRINT
460 PRINT S$(I)
470 PRINT
480 PRINT "WIE HEISST DIE AUTOMARKE?"
481 PRINT "HIER EINIGE VORSCHLÄGE"
482 PRINT
483 FOR G = 1 TO 6
484 LET H=6-G+1
485 PRINT L$(H)
486 NEXT G
```

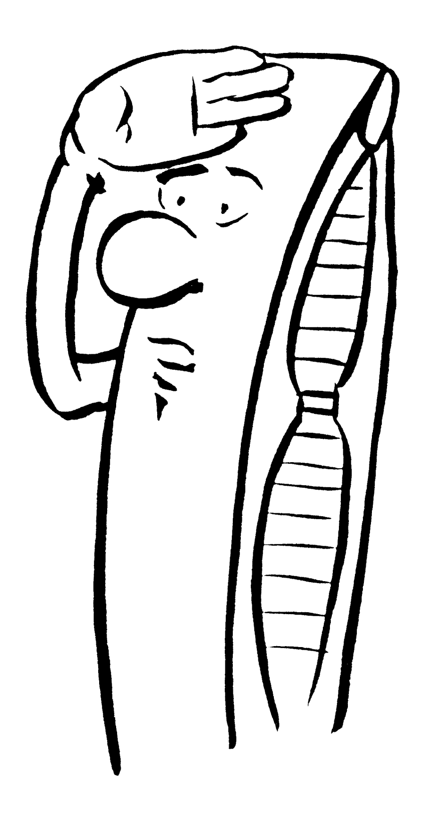
Fortsetzung auf der nächsten Seite ------------------------>

```
490 INPUT A$(I)
500 REM ******************************************************
510 REM *  3. Teil   -   Auswertung                          *
520 REM ******************************************************
530 IF A$(I)=L$(I) THEN GOTO 620
540 PRINT "DAS STIMMT LEIDER NICHT"
550 PRINT
560 PRINT "DIE RICHTIGE LOESUNG IST: ";
570 PRINT L$(I)
580 PRINT
590 PRINT "VERSUCHEN SIE ES NOCHMAL"
600 PRINT
610 GOTO 680
620 PRINT "TREFFER"
630 PRINT "GENAU DAS IST DIE LOESUNG"
640 PRINT
650 REM ******************************************************
660 REM *  4. Teil   -   Abfrage nochmal raten?              *
670 REM ******************************************************
680 PRINT "WOLLEN SIE NOCHMAL RATEN (JA/NEIN)?"
690 INPUT Q$
700 IF Q$="NEIN" THEN GOTO 770
710 NEXT I
720 PRINT "LEIDER NICHTS MEHR DA"
730 PRINT
740 REM ******************************************************
750 REM *  5. Teil   -   Gesamtliste                         *
760 REM ******************************************************
770 PRINT "WOLLEN SIE EINE LISTE"
780 PRINT "IHRER RATEVERSUCHE (JA/NEIN)?"
790 INPUT Q$
800 IF Q$="NEIN" THEN GOTO 900
810 PRINT
820 PRINT "SO HABEN SIE GERATEN"
830 PRINT
840 FOR J = 1 TO 6
850 PRINT S$(J);" - ";
860 PRINT A$(J);" (";
870 PRINT L$(J);")"
880 NEXT J
890 PRINT
900 PRINT "NUN IST SCHLUSS"
910 REM ******************************************************
920 REM *  Programmende                                      *
930 REM ******************************************************
```

Natürlich kennen Sie das Spiel „Ich denke mir eine Zahl zwischen . . ." Dieses Programm ist eine Computerversion davon. Die Maschine „denkt" sich eine Zahl und Sie müssen raten.

Knacken Sie die Zahlennuß

Für das Raten gibt es übrigens Strategien. Wir verraten Ihnen, wie Sie schnell zum Ziel kommen.



Zuerst aber wollen wir Ihnen das Programm erklären. Am Anfang haben wir einen kleinen Scherz eingebaut.

Der Computer fragt, ob Sie wirklich spielen wollen. Wenn Sie aber "NEIN" sagen, bricht er nicht ab, sondern läßt das Programm einfach weiterlaufen.

Zahl aus dem Generator

In Zeile 260 bekommt der Zufallszahlengenerator des Computers seinen Anfangswert. Natürlich sind die Zahlen nicht wirklich zufällig. Sie werden nach einer komplizierten Formel ausgerechnet.

Die „Zufälligkeit" der Zahlen hängt vom Anfangswert für die Formel ab. In unserem Programm bekommt der Computer eine Anfangszahl, die davon abhängt, wie lang der Computer schon eingeschaltet ist.

Ab Zeile 320 wird nun aktuell eine Zufallszahl bestimmt. Die Zahl Ihrer Rateversuche merkt sich der Computer unter X. X bekommt in Zeile 310 den Anfangswert 0.

Zufallszahlen aus dem Computer liegen zwischen 0 und 1. Damit die Zahl größer wird, multiplizieren wir sie in Zeile 330 mit 1000.

Der Computer meldet jetzt auf dem Bildschirm, daß er eine Zahl gefunden hat, und läßt Sie raten.

Bitte spielen Sie fair

Sie könnten natürlich das Programm kurz unterbrechen (mit BREAK), sich den Wert

von L ansehen und würden dann nach CONT im ersten Anlauf die richtige Zahl finden. Das ist aber nicht gerade die feine Art.

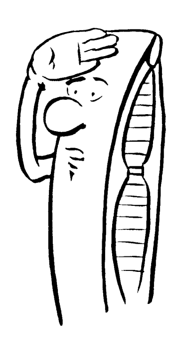
Raten Sie also einfach. Der Computer sagt Ihnen dann, ob Ihre Zahl zu groß, zu klein oder richtig ist.

Raten Sie mit Methode

Es gibt allerdings eine Möglichkeit, die Anzahl der Versuche möglichst gering zu halten.

Stellen Sie genau den Bereich fest, in dem Sie raten sollen. Am Anfang ist das 0 bis 1000. Versuchen Sie es dann immer mit der Zahl, die genau in der Mitte liegt. In diesem Fall ist das 500.

Sagt der Computer beispielsweise, daß die gesuchte Zahl größer ist, haben Sie schon die Hälfte aller Möglichkeiten ausgeschlossen. Jetzt wählen Sie – in unserem Beispiel – die Hälfte des Bereiches 500 bis 1000, also 750. Mit dieser Methode finden Sie schnell eine Lösung und verblüffen Ihre Freunde.



```
110 REM ***********************************************************
120 REM *  Nußknacker Programm   Zahlenraten    (45)              *
130 REM *                                                         *
140 REM *  1. Teil   -   Basisdaten   -   Initialisierung         *
150 REM ***********************************************************
160 PRINT "HALLO, HIER IST DAS"
170 PRINT "NUSSKNACKER PROGRAMM"
180 PRINT "MIT DEM SIE ZAHLEN"
190 PRINT "RATEN KOENNEN"
200 PRINT
210 PRINT "WOLLEN SIE (JA/NEIN)?"
220 INPUT A$
230 PRINT
240 IF A$="NEIN" THEN PRINT "PECH, SIE MUESSEN !!!"
250 PRINT
260 LET L=RND(-TI)
270 REM ***********************************************************
280 REM *  2. Teil   -   Raten                                    *
290 REM ***********************************************************
300 PRINT "ICH DENKE MIR JETZT EINE ZAHL"
310 LET X=0
320 LET L=RND(0)
330 LET L=L*1000
340 LET L=INT(L)
350 PRINT
360 PRINT "ICH HABE EINE ZAHL GEFUNDEN"
370 PRINT "SIE LIEGT ZWISCHEN 0 UND 1000"
380 PRINT
390 PRINT "VERSUCHEN SIE ES MAL"
400 INPUT P
410 PRINT
420 LET X=X+1
430 PRINT P
440 PRINT
450 REM ***********************************************************
460 REM *  3. Teil   -   Auswertung                               *
470 REM ***********************************************************
480 IF P=L THEN GOTO 540
490 IF P>L THEN GOTO 520
500 PRINT "IHRE ZAHL IST ZU KLEIN"
510 GOTO 380
520 PRINT "IHRE ZAHL IST ZU GROSS"
530 GOTO 380
540 PRINT "TREFFER"
550 PRINT "ANZAHL DER VERSUCHE: ";X
```

Die Untersuchung von Zahlenreihen liefert Ihnen hauptsächlich statistische Werte. Das kann aber ganz nützlich sein, wenn Sie etwa wissen wollen, wieviel Geld Sie durchschnittlich ausgeben, oder wie die Durchschnittsnote im Zeugnis aussieht.

Analyse von Zeugnis oder Budget

Also, ran an die Buletten, vertrauen Sie Ihre Zahlen dem Computer an.



Zuerst reservieren Sie im Speicher Platz für Ihre Zahlen. Sie werden unter W (für Werte) abgespeichert.

100 Werte sieht die DIM-Anweisung in Zeile 210 vor. Wollen Sie mehr als 100 Zahlen analysieren, setzten Sie den Wert entsprechend höher.

Vergabe der Anfangswerte

Basiswerte erhalten dann die Variablen N, X, D, P, S. Sie nehmen später das Minimum, das Maximum, die Differenz zwischen Minimum und Maximum, die entsprechende prozentuale Differenz und die Summe auf.

Die Anzahl der Werte, die Sie analysieren wollen, geben Sie in 280 an. Die Anzahl wird unter A gespeichert.

In der Schleife ab Zeile 360 fordert das Programm von Ihnen die Werte an und speichert sie gleich im Feld W. Diese Schleife wird so oft durchlaufen, wie Sie Werte eingeben wollen (A).

Auswertung Ihrer eingegebenen Werte

Sind alle Werte sicher „im Kasten“, beginnt die Auswertung Ihrer Zahlenreihe. In 510 beginnt eine Schleife, die so oft durchlaufen wird, wie Werte vorhanden sind. Die Schleife endet in Zeile 540.

Für jeden Wert bekommen Sie auf dem Bildschirm eine Zwischenauswertung. Der Bildschirm wird erst wieder gelöscht, wenn alle Ergebnisse für den nächsten Wert feststehen.

Besonders interessant ist natürlich, der

Durchschnittswert, der in 522 ausgerechnet und unter M gespeichert wird.

Sind alle Werte analysiert, und das kann bei vielen Werten eine ganze Weile dauern, gibt das Programm ab Zeile 624 die Schlußauswertung auf den Bildschirm.

Ist das Programm zu langsam?

Zur Beschleunigung des Programms können Sie übrigens die Zeilen 524 bis 539 weglassen. Sie bekommen dann die Zwischenauswertung nicht auf den Schirm. Dafür steht der Endwert schneller zur Verfügung.

Das können Sie übrigens auch elegant durch eine Abfrage am Anfang des Programms lösen, ob Zwischenwerte ausgedruckt werden sollen oder nicht. Abhängig von der Antwort überspringen Sie mit einem GOTO in der Schleife die Zeilen für die Ausgabe der Zwischenwerte.



```
110 REM ***********************************************************
120 REM *  Nußknacker Programm  Zahlenreihen    (414)             *
130 REM *                                                         *
140 REM *  1. Teil   -   Begrüßung   -    Variable                *
150 REM ***********************************************************
160 PRINT "HALLO, HIER IST EIN"
170 PRINT "NUSSKNACKER PROGRAMM"
175 PRINT
180 PRINT "SIE KOENNEN DAMIT"
190 PRINT "ZAHLENREIHEN UNTERSUCHEN"
200 PRINT
202 PRINT "TYPISCHE ZAHLENREIHEN SIND:"
203 PRINT
204 PRINT "BENZINVERBRAUCH PRO WOCHE"
205 PRINT "EINKUENFTE PRO MONAT"
206 PRINT "AUSGABEN PRO MONAT"
207 PRINT "NOTEN IM ZEUGNIS"
208 PRINT
210 DIM W(100)
212 LET N=65000
214 LET X=0
216 LET D=0
218 LET P=0
219 LET S=0
220 REM ***********************************************************
230 REM *  Anzahl der Werte                                       *
240 REM ***********************************************************
250 PRINT
260 PRINT "WIEVIEL WERTE WOLLEN SIE"
270 PRINT "EINGEBEN (MAX 100)?"
280 INPUT A
290 PRINT A
300 REM ***********************************************************
310 REM *  Eingabe der Werte                                      *
320 REM ***********************************************************
322 PRINT
324 PRINT
330 PRINT
340 PRINT "OK, GEBEN SIE JETZT DIE WERTE AN"
350 PRINT "VERGESSEN SIE NICHT: DER COMPUTER"
355 PRINT "WILL EINEN PUNKT STATT EINES KOMMAS"
360 FOR I = 1 TO A
370 PRINT
```

Fortsetzung auf der nächsten Seite ------------------------>

```
380 PRINT "WERT ";I;" ?"
390 PRINT
400 INPUT W(I)
410 PRINT
420 PRINT "WERT ";I;" IST ";W(I)
430 NEXT I
440 REM **********************************************************
450 REM *  Auswertung                                             *
460 REM **********************************************************
490 PRINT
500 LET S=0
510 FOR I=1 TO A
511 IF W(I)<N THEN LET N=W(I)
512 IF W(I)>X THEN LET X=W(I)
513 LET D=X-N
514 LET Z=X/100
516 LET P=N/Z
520 LET S=S+W(I)
522 LET M=S/I
523 PRINT
524 PRINT "ZWISCHENAUSWERTUNG"
526 PRINT
531 PRINT "NUMMER DES WERTES: ";I
532 PRINT
533 PRINT "WERT: ";W(I)
534 PRINT "SUMME DER WERTE: ";S
535 PRINT "DURCHSCHNITTSWERT: ";M
536 PRINT "KLEINSTER WERT: ";N
537 PRINT "GROESSTER WERT: ";X
538 PRINT "DIFFERENZ MIN/MAX: ";D
539 PRINT "PROZ. DIFFERENZ MIN/MAX: ";P
540 NEXT I
623 PRINT
624 PRINT "SCHLUSSAUSWERTUNG"
626 PRINT
631 PRINT "ANZAHL DER WERTE: ";A
634 PRINT "SUMME DER WERTE: ";S
635 PRINT "DURCHSCHNITTSWERT: ";M
636 PRINT "KLEINSTER WERT: ";N
637 PRINT "GROESSTER WERT: ";X
638 PRINT "DIFFERENZ MIN/MAX: ";D
639 PRINT "PROZ. DIFFERENZ MIN/MAX: ";P
700 REM **********************************************************
710 REM *  Programmende                                           *
720 REM **********************************************************
```

Sie sollten über längere Zeit den Durchschnittsverbrauch Ihres Autos beobachten. Daraus kann man viele Schlüsse auf Verschleiß und richtige Einstellung ziehen.

Die Liter fließen durch den Auspuff

Dieses Programm hilft Ihnen bei der Analyse. Geben Sie einfach die Werte ein. Den Durchschnitt berechnet der Computer.



Zuerst müssen Sie für die Analyse Platz im Speicher reservieren. Das passiert in Zeile 210. Das Feld W kann 100 Werte aufnehmen. Wenn Sie mehr Werte verarbeiten wollen, müssen Sie die Feldgröße entsprechend höher setzen. Sie legen damit nur die maximal mögliche Anzahl von Werten fest.

Wieviel Werte wollen Sie verarbeiten?

In Zeile 280 können Sie dann angeben, wieviel Werte sie aktuell eingeben wollen. Dabei dürfen Sie den in 210 angegebenen Wert nicht übersteigen.

In der Schleife ab Zeile 360 beginnt dann die Eingabe der Werte. Ihre Angabe, wieviel Werte Sie analysieren wollen, ist ja aus Zeile 280 in A gespeichert.

Dieses A bestimmt dann in Zeile 360, wie oft die Schleife durchlaufen werden muß, damit Sie alle Werte eingeben können.

Das Programm nennt Ihnen bei jeder Eingabe die Nummer des nächsten Wertes und wiederholt den Wert auf dem Bildschirm.

Ein Verbrauchswert pro Feldelement

Jedes Element des Feldes W entspricht einem von Ihnen eingegebenen Wert. Im ersten Element steht der erste Wert, im sechsten der sechste usw.

Wenn alle Werte eingegeben sind, wenn also die Schleife A-mal durchlaufen wurde, beginnt die Auswertung bei Zeile 480.

Dazu erhält zuerst die Variable, die die Summe aller von Ihnen eingegebenen Werte

aufnehmen soll (S wie Summe) als Anfangswert eine 0.

Dann beginnt wieder eine Schleife, die so oft durchlaufen wird, wie Werte vorhanden sind. (Hier wirkt immer noch Ihre Eingabe aus Zeile 280.)

Jetzt haben wir gleich den Durchschnitt

Bei jedem Durchlauf wird ein von Ihnen eingegebener Wert zu der Summe der bisher zusammengezählten Werte addiert (Zeile 520). Das Ergebnis muß man dann nur noch durch die Anzahl der Werte dividieren und schon steht der Durchschnittsverbrauch fest.

Natürlich können Sie mit diesem Programm nicht nur den Durchschnittsverbrauch Ihres Autos feststellen, sondern den Durchschnittswert jeder beliebigen Zahlenreihe.

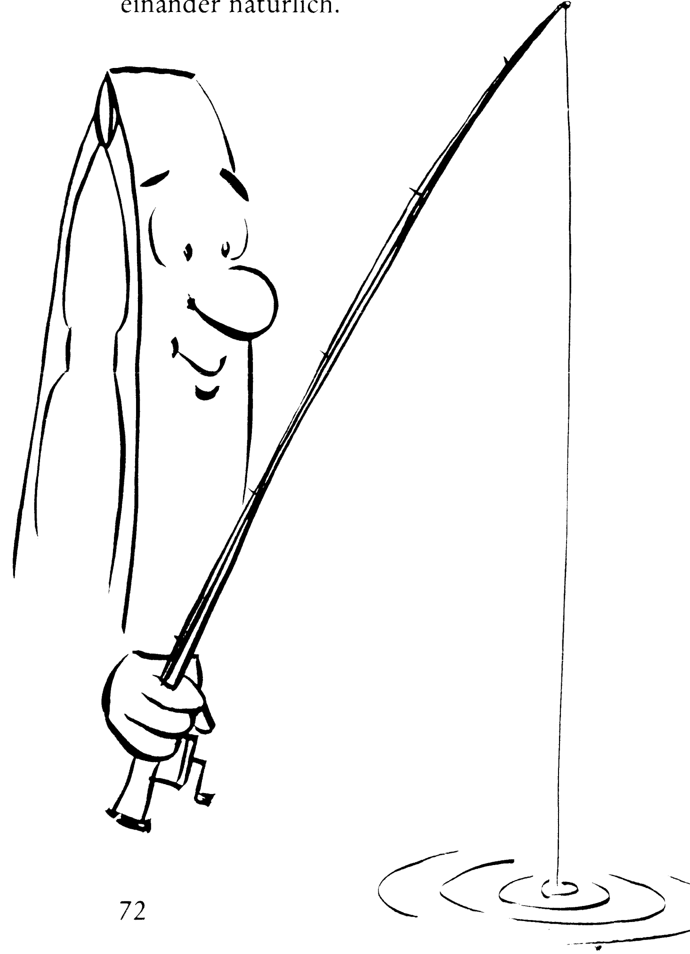


```
110 REM ******************************************************
120 REM *  Nußknacker Programm  Durchschnittsverbrauch (410) *
130 REM *                                                    *
140 REM *  1. Teil    -    Begrüßung    -    Speicher        *
150 REM ******************************************************
160 PRINT "HALLO, HIER IST EIN"
170 PRINT "NUSSKNACKER PROGRAMM"
180 PRINT "SIE KOENNEN DAMIT DEN"
190 PRINT "DURCHSCHNITTSVERBRAUCH"
200 PRINT "IHRES AUTOS AUSRECHNEN"
210 DIM W(100)
220 REM ******************************************************
230 REM *  Anzahl der Werte                                  *
240 REM ******************************************************
250 PRINT
260 PRINT "WIEVIEL WERTE WOLLEN SIE"
270 PRINT "EINGEBEN (MAX 100)?"
280 INPUT A
290 PRINT A
300 REM ******************************************************
310 REM *  Eingabe der Werte                                 *
320 REM ******************************************************
330 PRINT
340 PRINT "OK, GEBEN SIE JETZT DIE WERTE AN"
350 PRINT "VERBRAUCH IN LITERN"
355 PRINT "PRO 100 KM" (PUNKT STATT KOMMA)"
360 FOR I = 1 TO A
370 PRINT
380 PRINT "WERT ";I;" ?"
390 PRINT
400 INPUT W(I)
410 PRINT
420 PRINT "WERT ";I;" IST ";W(I)
430 NEXT I
440 REM ******************************************************
450 REM *  Auswertung                                        *
460 REM ******************************************************
470 PRINT
480 PRINT "DIE AUSWERTUNG:"
490 PRINT
500 LET S=0
510 FOR I=1 TO A
520 LET S=S+W(I)
530 NEXT I
540 PRINT "GESAMT: ";S;" L *** DURCHSCHNITT: ";S/A;" L"
```

Trainieren Sie Ihre kleinen grauen Gehirnzellen mit diesem Programm. Wenn Sie sich wirklich anstrengen, kommen Sie ganz schön in's Schwitzen.

Angeln Sie nach der richtigen Zahl

In den drei Grundrechenarten stellt Ihnen der Computer immer neue Fragen. Durcheinander natürlich.



Zuerst können Sie selbst den Schwierigkeitsgrad für die Rechnerei festlegen. Am Anfang sollten Sie sich mit kleinen Zahlen begnügen.

Das ist anstrengend genug. Später, wenn Sie mit den kleinen Zahlen gut klar kommen, können Sie auch im Schwierigkeitsgrad weiter nach oben klettern.

Das ist anstrengender als laufen

Oder, wenn Sie sich zu regelmäßigem Kopfrechnen durchringen können, steigern Sie einfach den Schwierigkeitsgrad täglich um 1. Eine Stunde Kopfrechnen ist übrigens mindestens so anstrengend wie ein 1000-Meter-Lauf. Viel Erfolg.

Falls Sie den Schwierigkeitsgrad nicht innerhalb der erlaubten Grenzen angeben, müssen Sie die Eingabe wiederholen.

Beginnen Sie mit 10

Eine Anfangszahl der Rechnerei ist immer 10 (Z). Mit A zählt man die Anzahl der Rechenoperationen. Der Wert von A wird ausgegeben, wenn Sie sich verrechnen. Sie bekommen dann die Anzahl der richtig ausgeführten Rechnungen genannt.

Führen Sie Buch über Ihre Erfolge

Stellen Sie Ihren persönlichen Rekord auf und notieren Sie sich dafür Schwierigkeitsgrad und erreichte Punktzahl.

Ab Zeile 320 wird dann zufällig eine Zahl für die weitere Rechnerei gewählt und mit dem Schwierigkeitsgrad multipliziert.

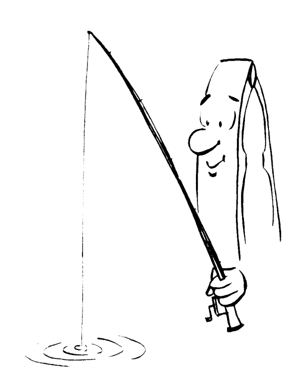
Die präsentiert Ihnen dann der Computer und läßt Sie damit Addieren, Subtrahieren oder Multiplizieren.

Die Rechenart bestimmt der Computer

Die Rechenart hängt mit der Größe der bisher errechneten Zahl zusammen. Das Programm sorgt dafür, daß sie nicht zu groß und nicht zu klein wird.

Multipliziert wird in Zeile 430, addiert in Zeile 460 und subtrahiert in Zeile 490. Ihre Eingabe wird dann in Zeile 505 angefordert.

In Zeile 510 wird Ihre Lösung geprüft. War Sie richtig, bekommen Sie eine neue Aufgabe, war sie falsch, nennt der Computer Ihnen die errechnete Punktezahl.



```
100 REM ***************************************************
120 REM *  Nußknacker Programm   Kopfrechnen    (47)      *
130 REM *                                                 *
140 REM *  1. Teil   -   Begrüßung   -   Schwierigkeit    *
150 REM ***************************************************
160 PRINT "HALLO, HIER IST DAS"
170 PRINT "NUSSKNACKER PROGRAMM"
180 PRINT "ZUM TRAINIEREN DES"
190 PRINT "* KOPFRECHNENS *"
200 PRINT
210 PRINT "SCHWIERIGKEITSGRAD (10 bis 1000)?"
220 INPUT S
230 IF S>9 AND S<1001 THEN GOTO 300
240 PRINT "FALSCHE EINGABE"
250 PRINT "VERSUCHEN SIE ES NOCHMAL"
260 GOTO 200
270 REM ***************************************************
280 REM *  2. Teil   -   Rechnen                          *
290 REM ***************************************************
300 N=RND(-TI)
310 LET Z=10
320 LET A=0
330 LET N=RND(0)
340 LET N=N*S
350 LET N=INT(N)
360 LET A=A+1
370 PRINT
380 PRINT "DIE ZAHL IST: ";Z
390 PRINT
400 IF Z<=S THEN GOTO 450
410 IF Z>S*100 THEN GOTO 480
420 PRINT "MULTIPLIZIEREN SIE MIT ";N
430 LET Z=Z*N
440 GOTO 500
450 PRINT "ADDIEREN SIE ";N
460 LET Z=Z+N
470 GOTO 500
480 PRINT "SUBTRAHIEREN SIE ";N
490 LET Z=Z-N
500 PRINT
505 INPUT X
510 IF X=Z THEN GOTO 330
515 PRINT
520 PRINT "FALSCH ";
530 PRINT "RICHTIG GELOESTE AUFGABEN: ";A-1
```

Wenn Sie viel mit Mehrwertsteuer zu tun haben, kennen Sie die Notwendigkeit, den MwSt.-Anteil aus einem Bruttobetrag herauszurechnen und dann noch längere Listen zu addieren.

Vorsteuer, kein Problem

Genau dieses Problem löst für Sie unser Programm. Vertrauen Sie ihm getrost die Rechnerei an.



Selten wird bei kleinen Beträgen auf der Rechnung der Mehrwertsteueranteil gesondert ausgewiesen. Da steht dann lapidar: „Inklusive soundsoviel Prozent Mehrwertsteuer".

Was tut man nicht für's Finanzamt!

Für die Buchhaltung und für's Finanzamt braucht man dann aber doch Brutto- und Nettobetrag sowie MwSt.-Anteil gesondert.
Genau diese Rechnerei nimmt Ihnen unser Programm ab. Wenn Sie gleichzeitig noch mehrere Beträge aufaddieren wollen, das schafft das Programm ganz nebenbei.

Ein Drucker schafft Belege

Besonders sinnvoll ist es, einen Drucker anzuschließen. Sie haben dann gleich einen Beleg, wenn Sie die Ausgabeanweisung von „Bildschirm" auf „Drucker" ändern. Oder Sie lassen beides parallel laufen.
Das Programm verlangt von Ihnen zuerst die Angabe des Mehrwertsteuersatzes in Prozent. Den geben Sie über die Tastatur ein.

Geben Sie jetzt die Bruttowerte ein

Jetzt können Sie die eigentlichen Werte eingeben. Von Ihnen kommen über die Tastatur die Bruttowerte.
Der Computer rechnet den Nettobetrag aus und anschließend den Mehrwertsteueranteil. Komplett erhalten Sie auf dem Bild-

schirm Originalwert, Nettowert und den Mehrwertsteueranteil.

Addiert wird gleich mit

Gleichzeitig wird summiert. Alle von Ihnen eingegebenen Werte addiert der Computer. Neben der Bruttosumme erhalten Sie auch die Angaben wie beim Einzelwert.

Wenn Sie alle Zahlen eingegeben haben, können Sie das Programm durch Eingabe von 99999 beenden. Wir wünschen lustiges Rechnen!



```
110 REM ***********************************************************
120 REM *  Nußknacker Programm   Anteil Mwst.   (415)             *
130 REM *                                                         *
140 REM *  1. Teil   -   Basisdaten   -   Mwst-Satz               *
150 REM ***********************************************************
160 PRINT "HALLO, HIER IST DAS"
170 PRINT "NUSSKNACKER PROGRAMM"
180 PRINT "DASS GANZE ZAHLENKOLONNEN"
190 PRINT "ADDIERT UND DABEI DIE"
200 PRINT "MEHRWERTSTEUER AUSWIRFT"
210 PRINT
220 PRINT "EINGABE BRUTTOWERTE"
240 PRINT "PROGRAMMENDE MIT 99999"
250 PRINT
260 PRINT "WIE IST DER MWST-SATZ"
270 PRINT "IN PROZENT ?"
280 INPUT M
290 LET A=0
300 LET S=0
310 REM ***********************************************************
320 REM *  2. Teil   -   Eingabe der Werte/Berechnung             *
330 REM ***********************************************************
340 LET A=A+1
350 PRINT "GEBEN SIE BRUTTOWERT ";A;" EIN"
360 INPUT W
370 IF W=99999 THEN GOTO 550
380 PRINT
390 PRINT "NUMMER DES WERTES: ";A
400 PRINT
410 PRINT "BRUTTOWERT: ";W
420 LET N=W/(1+M/100)
430 PRINT "NETTOWERT": ";N
440 LET X=W-N
450 PRINT "MEHRWERTSTEUER-ANTEIL: ";X
460 LET S=S+W
470 LET W=S
475 PRINT
480 PRINT "SUMME BRUTTOWERT: ";W
490 LET N=W/(1+M/100)
500 PRINT "DAVON NETTOWERT": ";N
510 LET X=W-N
520 PRINT "DAVON MEHRWERTSTEUER-ANTEIL: ";X
530 PRINT
540 GOTO 380
550 REM ***********************************************************
```

Wenn Sie selbst mehrwertsteuerpflichtig sind, müssen Sie zu jedem Betrag die Mehrwertsteuer ausrechnen und sie zum Nettobetrag addieren. Das hält ganz schön auf, besonders wenn gleichzeitig die Beträge zu addieren sind.

Mehrwertsteuer, kein Problem

Genau dieses Problem löst für Sie unser Programm. Vertrauen Sie ihm getrost die Rechnerei an.



Ganz schön lästig ist die Mehrwertsteuer, besonders, weil man nichts davon hat als Arbeit. Das Geld bekommt doch das Finanzamt. Interessant ist eigentlich nur der Nettobetrag.

Dagegen ist kein Kraut gewachsen

Doch was hilft alles Lamentieren. Die Rechnerei muß sein, ob man nun Lust hat oder auch nicht.

Genau diese Rechnerei nimmt Ihnen unser Programm ab. Wenn Sie gleichzeitig noch mehrere Beträge aufaddieren wollen, das schafft das Programm ganz nebenbei.

Ein Drucker schafft Belege

Besonders sinnvoll ist es, einen Drucker anzuschließen. Sie haben dann gleich einen Beleg, wenn Sie die Ausgabeanweisung von „Bildschirm" auf „Drucker" ändern. Oder Sie lassen beides parallel laufen.

Das Programm verlangt von Ihnen zuerst die Angabe des Mehrwertsteuersatzes in Prozent. Den geben Sie über die Tastatur ein.

Geben Sie jetzt die Nettowerte ein

Jetzt können Sie die eigentlichen Werte eingeben. Von Ihnen kommen über die Tastatur die Nettowerte.

Der Computer rechnet den Bruttobetrag aus und anschließend die Mehrwertsteuer. Komplett erhalten Sie auf dem Bildschirm

Originalwert, Bruttowert und die Mehrwertsteuer.

Addiert wird gleich mit

Gleichzeitig wird summiert. Alle von Ihnen eingegebenen Werte addiert der Computer. Neben der Nettosumme erhalten Sie auch die Angaben wie beim Einzelwert.

Wenn Sie alle Zahlen eingegeben haben, können Sie das Programm durch Eingabe von 99999 beenden. Wir wünschen lustiges Rechnen!

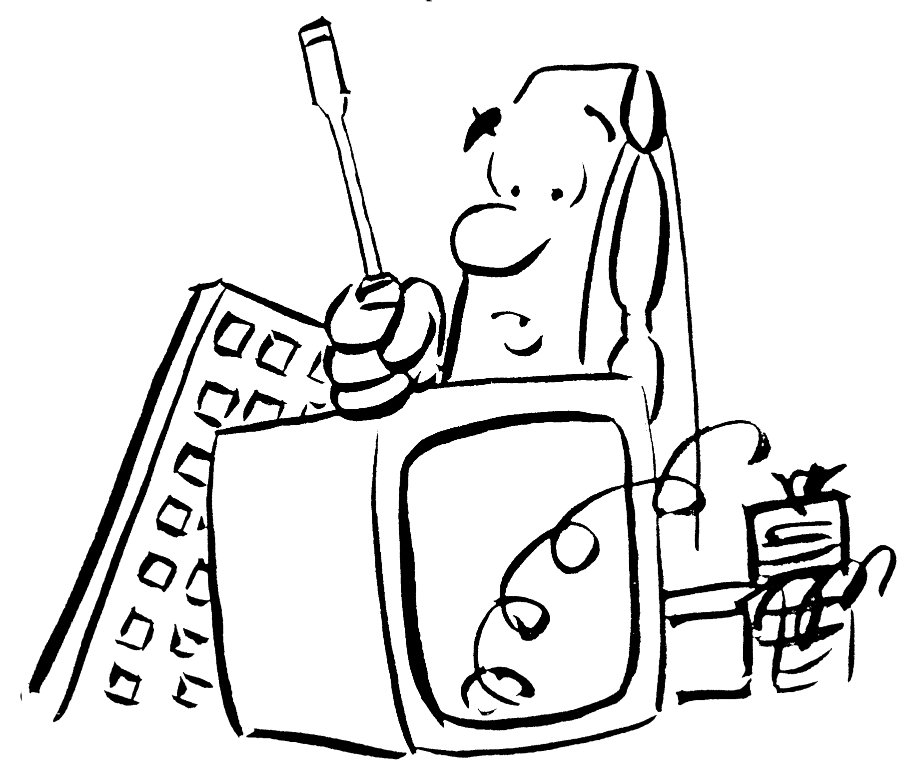


```
110 REM **********************************************************
120 REM *  Nußknacker Programm   Mehrwertsteuer (416)            *
130 REM *                                                        *
140 REM *  1. Teil   -    Basisdaten   -   Mwst-Satz             *
150 REM **********************************************************
160 PRINT "HALLO, HIER IST DAS"
170 PRINT "NUSSKNACKER PROGRAMM"
180 PRINT "DASS GANZE ZAHLENKOLONNEN"
190 PRINT "ADDIERT UND DABEI DIE"
200 PRINT "MEHRWERTSTEUER AUSWIRFT"
210 PRINT
220 PRINT "EINGABE BRUTTOWERTE"
240 PRINT "PROGRAMMENDE MIT 99999"
250 PRINT
260 PRINT "WIE IST DER MWST-SATZ"
270 PRINT "IN PROZENT ?"
280 INPUT M
290 LET A=0
300 LET S=0
310 REM **********************************************************
320 REM *  2. Teil   -    Eingabe der Werte/Berechnung           *
330 REM **********************************************************
340 LET A=A+1
350 PRINT "GEBEN SIE NETTOWERT ";A;" EIN"
360 INPUT W
370 IF W=99999 THEN GOTO 550
380 PRINT
390 PRINT "NUMMER DES WERTES: ";A
400 PRINT
410 PRINT "NETTOWERT: ";W
420 LET N=W*(1+M/100)
430 PRINT "BRUTTOWERT": ";N
440 LET X=N-W
450 PRINT "MEHRWERTSTEUER: ";X
460 LET S=S+W
470 LET W=S
475 PRINT
480 PRINT "SUMME NETTOWERT: ";W
490 LET N=W*(1+M/100)
500 PRINT "DAZU BRUTTOWERT": ";N
510 LET X=N-W
520 PRINT "DAZU MEHRWERTSTEUER: ";X
530 PRINT
540 GOTO 380
550 REM **********************************************************
```

Einen elektronischen Würfelbecher haben Sie mit diesem Programm. Maximal acht Würfel passen hinein und natürlich können Sie jedes Würfelauge auf dem Bildschirm sehen.

Elektronischer Würfelbecher

Austricksen können Sie Ihre Mitspieler mit diesem Programm nicht. Und wenn Sie wirklich mal ein paar Sechser haben? Zufall!



Zufallszahlen aus dem Computer sind übrigens nicht wirklich zufällig. Sie werden nach einer komplizierten Formel berechnet, der man einen Anfangswert geben muß. Danach kommen die Zahlen ziemlich durcheinander aus dem Computer gepurzelt.

Geben Sie also zweimal die gleiche Zahl als Anfangswert an, so bekommen Sie später auch die gleichen „zufälligen" Zahlen. Das Problem ist also, immer wieder unterschiedliche und „zufällige" Anfangszahlen zu finden.

Zufall hängt von der Laufzeit ab

Das ist in Zeile 260 so gelöst, daß der Computer auf eine Speicherzelle zurückgreift, die abhängig von der bisherigen Laufzeit der Maschine ist.

Damit haben Sie also den Zufall in Ihren elektronischen Würfelbecher einprogrammiert. Vorher legen Sie mit der INPUT-Anweisung in Zeile 240 noch die Anzahl der Würfel fest, die sich der Computer unter WA (Würfelanzahl) merkt.

Strenge Kontrollen bei der Würfelanzahl

Dann brauchen Sie Platz auf dem Bildschirm (Zeile 300) und schon rollen die Würfel. Die in Zeile 330 beginnende Schleife wird so oft durchlaufen, wie Würfel im elektronischen Würfelbecher sind (WA).

Damit diese Schleife übrigens nicht zu oft durchlaufen wird, oder falls sogar ein Narr eine negative Würfelanzahl eingegeben hat, prüft das Programm in Zeile 250 genau, ob die

Anzahl der Würfel mit den Spielregeln übereinstimmt.

In unserer Schleife ab 330 fordert das Programm jetzt eine Zufallszahl an. Diese Zahlen liegen immer zwischen 0 und 1. Um also eine Zahl im richtigen Bereich zu bekommen, muß man mit 10 multiplizieren.

Würfeln hinter dem Komma?

Die Stellen hinter dem Komma interessieren nicht, oder haben Sie schon einmal einen Würfel mit 2.34 Augen gesehen? Alle Zahlen hinter dem Komma werden durch INT abgeschnitten.

Jetzt muß man nur noch feststellen, ob die Zufallszahl wirklich für das Würfeln in Frage kommt. Schließlich interessieren da nur Zahlen zwischen 1 und 6.

Diese Prüfung erfolgt in Zeile 370. Falls die Zahl unseren Anforderungen nicht entspricht, vergessen wir sie einfach und fordern in Zeile 340 eine neue an.

Für jedes „Auge" ein Stern

Ist die Zahl jedoch akzeptabel, wird sie in der Schleife ab Zeile 410 bildlich dargestellt. Für jedes Auge erscheint ein * auf dem Bildschirm.

Noch ist die in Zeile 310 begonnene Schleife nicht beendet. Sind mehrere Würfel im elektronischen Würfelbecher, fällt jetzt der nächste Würfel aus dem Becher.

Interessant ist übrigens noch ein Blick auf das Semikolon in Zeile 420. Es verhindert,

daß die einzelnen Sterne untereinander stehen. Damit aber die Sternchen des nächsten Durchgangs nicht auch noch neben die „Augen“ dieses Durchgangs geraten, gehen wir mit der PRINT-Anweisung in Zeile 440 in die nächste Zeile. Es folgt noch eine Leerzeile (450).

Mit NEXT A in 460 ist dann der Durchgang für einen Würfel beendet. Sind mehrere Würfel da, durchläuft das Programm die Schleife, die in 330 begonnen hat, noch einmal.



Landet das Programm in der Zeile 470, sind die Würfel gefallen. Für jeden Würfel im Becher sind die „Augen“ in einer Zeile auf dem Bildschirm.

Wieviel Würfel in der neuen Runde?

Da bleibt dem Programm nur noch festzustellen, ob noch weiterer Bedarf besteht. Sagen Sie JA, wenn Sie noch weiter spielen wollen. Geben Sie etwas Anderes als JA ein, schmollt der Computer mit ”DANN EBEN NICHT“.

Falls Sie vor jedem neuen Wurf die Anzahl der Würfel neu bestimmen wollen, sollten Sie in Zeile 520 die 300 in eine 210 ändern. Der Computer fragt Sie dann, mit wieviel Würfeln Sie spielen wollen.

Lassen Sie das Programm aber unverändert, können Sie die Zahl der Würfel nicht mehr ändern, wenn sie einmal festliegt.

```
110 REM *******************************************************
120 REM *  Nußknacker Programm   Würfelbecher   (42)          *
130 REM *                                                     *
140 REM *  1. Teil   -   Basisdaten   -   Initialisierung     *
150 REM *******************************************************
160 PRINT "HALLO, HIER IST DAS"
170 PRINT "NUSSKNACKER PROGRAMM"
180 PRINT "MIT IHREM PERSOENLICHEN"
190 PRINT "WUERFELBECHER"
200 PRINT
210 PRINT "MIT WIEVIEL WUERFELN"
220 PRINT "WOLLEN SIE SPIELEN?"
230 PRINT "MAXIMAL ACHT STUECK!"
240 INPUT WA
250 IF WA>8 OR WA<1 THEN GOTO 230
260 LET L=RND(-TI)
270 REM *******************************************************
280 REM *  2. Teil   -  Die Würfel rollen                     *
290 REM *******************************************************
300 PRINT
310 PRINT "DIE WUERFEL ROLLEN"
320 PRINT
330 FOR A = 1 TO WA
340 LET L=RND(0)
350 LET L=L*10
360 LET L=INT(L)
370 IF L>6 OR L<1 THEN GOTO 340
380 REM *******************************************************
390 REM *  3. Teil   -  Auf den Bildschirm                    *
400 REM *******************************************************
410 FOR I = 1 TO L
420 PRINT "* ";
430 NEXT I
440 PRINT
450 PRINT
460 NEXT A
470 REM *******************************************************
480 REM *  4. Teil   -   Wiederholung oder Ende               *
490 REM *******************************************************
500 PRINT "WOLLEN SIE NOCHMAL WUERFELN (JA/NEIN)?"
510 INPUT A$
520 IF A$="JA" THEN GOTO 300
530 PRINT
540 PRINT "DANN EBEN NICHT"
```

Wer möchte nicht gerne im Lotto gewinnen? Ob Sie in Zukunft mit diesem Programm mehr Glück haben werden, wissen wir nicht, aber wir hoffen es für Sie.

Machen Sie keinen Fehler

Allerdings benutzen wir dieses Programm auch und hoffen für uns noch mehr. Wem mag die Glücksfee hold sein?



Mit Sicherheit machen Sie einen Fehler, wenn Sie glauben, der Computer könne Ihnen die kommenden Lottozahlen ausrechnen. Die Wahrscheinlichkeit spricht dagegen.

Ein Fehler ist es aber nicht, den Computer so viele Tips ausrechnen zu lassen, bis einer dabei ist der Ihnen gefällt.

Ein neuer Tip für jede Woche

So haben Sie auch jede Woche einen neuen Tip und brauchen keine Angst zu haben, daß Sie mal den Lottoschein abzugeben vergessen haben. Denn das kann nur dann ärgerlich sein, wenn Sie immer den gleichen Tip benutzen.

Im Programm wird anfangs Ihre persönliche Glückszahl ausgerechnet. Dazu benutzt der Computer Ihren Geburtstag und das Tagesdatum.

Im ersten Teil fordert er dazu die Daten an, im zweiten Teil berechnet er dann die Zahl ab Zeile 340.

Der Anfang für den Zufall

Mit dieser Zahl bekommt der Computer einen Anfangswert für den Zufallszahlengenerator. Den braucht er, damit er immer unterschiedliche Zahlen generiert.

Für die Lottozahlen wird dann Platz im Speicher reserviert. Sechs Zahlen können Sie in einem Feld unterbringen und den Platz dafür haben Sie im Feld Z.

In der in Zeile 450 beginnenden Schleife

werden dann die Zufallszahlen auf Ihre Lotto-Tauglichkeit hin untersucht.

Die Schleife wird sechsmal durchlaufen, einmal für jede Zahl eines Feldes. Da die Zufallszahlen zwischen 0 und 1 liegen, müssen Sie zuerst mit 100 multipliziert werden (Zeile 470), um überhaupt in den Lotto-Bereich zu kommen.

Die Guten in's Töpfchen, die Schlechten . . .

Erlaubt sind aber nur Zahlen zwischen 1 und 49. Das wird in Zeile 500 geprüft. Unpassende Zahlen vergißt der Computer einfach und holt sich ab Zeile 460 eine neue Zahl. Ist die Zahl aber im Lottobereich, wird Sie vorläufig akzeptiert.

Keine zwei Kreuze auf einem Feld

Sie hat allerdings noch eine Prüfung zu bestehen. In der Schleife ab Zeile 510 wird Sie mit allen bereits akzeptierten Zahlen verglichen. Kam die Zahl schon einmal vor, vergißt der Computer sie ebenfalls und macht einen neuen Versuch. Schließlich können Sie auf dem Lottoschein ja nicht zwei Kreuze übereinander machen.



Sind schließlich sechs Zahlen gefunden, schreibt der Computer sie in der ab Zeile 630 beginnenden Schleife auf den Bildschirm.

Mit der Eingabe in Zeile 720 können Sie einen neuen Tip anfordern oder das Programm beenden.

```
100 REM ***********************************************************
110 REM * Nußknacker Programm   LOTTOZAHLEN   (41)                *
120 REM *                                                         *
130 REM * 1. Teil    -    Basisdaten                              *
140 REM ***********************************************************
150 PRINT "HALLO, HIER IST DAS"
160 PRINT "NUSSKNACKER PROGRAMM"
170 PRINT "FUER IHRE PERSOENLICHEN"
180 PRINT "LOTTOZAHLEN"
190 PRINT
200 PRINT "ZUERST ERRECHNEN WIR IHRE"
210 PRINT "PERSOENLICHE GLUECKSZAHL"
220 PRINT
230 PRINT "GEBEN SIE IHR GEBURTSJAHR AN"
240 INPUT GJ
250 PRINT "GEBEN SIE IHREN GEBURTSTAG AN (1-31)"
260 INPUT GT
270 PRINT "WELCHEN MONAT HABEN WIR HEUTE (1-12)?"
280 INPUT MO
290 PRINT "DER WIEVIELTE IST HEUTE (1-31)?"
300 INPUT TA
310 REM ***********************************************************
320 REM * 2. Teil    -    persönliche Glückszahl                  *
330 REM ***********************************************************
340 LET GZ=GT-TA
350 LET GZ=GZ*GJ
360 LET GZ=GZ/MO
370 LET GZ=ABS(GZ)
380 PRINT
390 PRINT "IHRE GLUECKSZAHL IST: ";GZ
400 L=RND(-GZ)
410 REM ***********************************************************
420 REM * 3. Teil    -    Ausrechnen der Lottozahlen              *
430 REM ***********************************************************
440 DIM Z(6)
450 FOR I = 1 TO 6
460 LET L=RND(0)
470 LET L=L*100
480 LET L=INT(L)
490 PRINT L
500 IF L>49 OR L<1 THEN GOTO 460
```

Fortsetzung auf der nächsten Seite ------------------------>

```
510 FOR J = 1 TO I
520 IF L=Z(J) THEN GOTO 460
530 NEXT J
540 Z(I)=L
550 NEXT I
560 REM ******************************************************
570 REM * 4. Teil   -   Ausgabe der Lottozahlen              *
580 REM ******************************************************
590 PRINT
600 PRINT
610 PRINT "IHR PERSOENLICHER LOTTOTIP:"
620 PRINT
630 FOR K = 1 TO 6
640 PRINT Z(K);" ";
650 NEXT K
660 PRINT
670 PRINT
680 REM ******************************************************
690 REM * 5. Teil   -   Wiederholung oder Ende ?             *
700 REM ******************************************************
710 PRINT "WOLLEN SIE NOCH EINEN TIP (JA/NEIN)?"
720 INPUT A$
730 IF A$="JA" THEN GOTO 450
740 PRINT
750 PRINT "DANN EBEN NICHT"
760 REM ******************************************************
770 REM * Programmende                                       *
780 REM ******************************************************
```

```
IHR PERSOENLICHER LOTTOTIP:

31 12 6 39 4 17
```

Diesen Tip haben wir erhalten, wie sieht Ihrer aus ?

BASIC zum Nachschlagen



Hier haben wir noch einmal alle Schlüsselworte des Commodore-64-Basic in alphabetischer Reihenfolge zusammengestellt und mit kurzen Erklärungen und kleinen Beispielen versehen. Sie können in diesem Teil schnell mal nachschlagen, wenn Ihnen die Bedeutung eines Befehls nicht geläufig ist.

ABS — das immer positive Ergebnis

Mit dieser Funktion können Sie den Computer den Absolutbetrag eines Wertes ausrechnen lassen. Das Ergebnis ist immer eine positive Zahl. Der Absolutbetrag von 0 ist 0.

Beispiel: Die Anweisungen PRINT ABS (3) und PRINT ABS (−3) bringen als Ergebnis jeweils eine 3 auf den Bildschirm.

AND — das logische UND

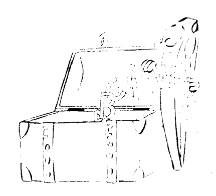
Die wichtigste Aufgabe erledigt das Schlüsselwort AND in der IF-Anweisung. Da stellen Sie ja dem Computer eine Bedingung. Wenn die Bedingung erfüllt ist,dann tue das und das. Nun kann es aber auch notwendig sein, die Ausführung einer Anweisung an mehrere Bedingungen zu knüpfen. Sie soll nur dann ausgeführt werden, wenn alle Bedingungen erfüllt sind. Dafür gibt es das Schlüsselwort AND.

. . . dann und nur dann

Beispiel: IF A > 10 AND A < 15 THEN PRINT A. Bei dieser Anweisung bekommen Sie den Wert von A nur auf den Bildschirm,

wenn er größer als 10 und kleiner als 15 ist. Der Befehl PRINT A wird also nur dann ausgeführt, wenn A einen Wert zwischen 11 und 14 hat.

ASC — welche Nummer hat der Buchstabe?



Wir haben schon den Befehl CHR$ kennengelernt. ASC ist die Umkehrung davon. Das Argument für die Funktion ASC ist eine Zeichenfolge. Das Ergebnis ist die Nummer des ersten Zeichens. Wenn das erste Zeichen also beispielsweise ein A ist, liefert ASC als Ergebnis 65.

Beispiel: Die Anweisung PRINT ASC ("ABCD") schreibt auf den Bildschirm die Zahl 65. PRINT ASC ("") schreibt eine 0 auf den Bildschirm.

ATN — der Arcustangens

Mit dieser Funktion können Sie den Computer den Arcustangens in Bogengraden ausrechnen lassen.

Beispiel: Die Anweisung PRINT ATN (−0.5) bringt richtig den Wert 0.46364761 auf den Bildschirm.

CHR$ — Buchstaben aus Zahlen

Jedes Zeichen kennt der Computer nur unter einer Nummer. A hat beispielsweise die Nummer 65 und B die Nummer 66. Mit der Funktion CHR$ können Sie diese Nummer in das Zeichen verwandeln.

Beispiel: PRINT CHR$ (65) bringt auf den Bildschirm das Zeichen A. Mit der Anweisung LET X$ = CHR$ (65) weisen Sie X$ das Zeichen "A" zu.

CLOSE — machen Sie dicht

Mit CLOSE können Sie geöffnete Files wieder schließen. Erst dann sind sie richtig gespeichert. Hinter CLOSE können Sie die Nummer des Files angeben.

Beispiel: CLOSE 3 schließt das vorher unter Nr. 3 geöffnete File.

CLR — schafft Platz im Speicher

Mit der Anweisung CLR löschen Sie im Speicher alle Bereiche, in denen Werte aufbewahrt sind. Das Programm bleibt — anders als bei NEW — erhalten. CLR benutzt man dann, wenn im Speicher nicht mehr genug Platz ist.

. . . und Computer vergessen doch

Beispiel: Mit der Anweisung LET A = 10 weisen Sie der Variablen A den Wert 10 zu. Nach der Ausführung des Befehls CLR hat der Computer diesen Befehl wieder „vergessen“.

CMD — Umleitung zur Peripherie

CMD leitet die mit LIST oder PRINT erzeugten Ausgaben auf ein anderes Gerät um als den Bildschirm. Sie müssen es vorher mit

OPEN öffnen. Hinter CMD geben Sie das Gerät oder File an.

Beispiel: CMD 1 leitet die von LIST oder PRINT erzeugten Ausgaben auf File 1 oder Gerät 1.

CONT — Fortsetzung folgt

CONT ist eine Abkürzung für CONTINUE, was etwa WEITERMACHEN heißt. Diesen Befehl können Sie immer dann benutzen, wenn der Computer stehengeblieben ist. Das passiert zum Beispiel, wenn eine STOP-Anweisung im Programm steht. Befehlen Sie dann dem Computer mit CONT an der Stelle weiterzumachen, wo er die Programmausführung unterbrochen hat.

COS — der Cosinus

Mit dieser Funktion können Sie den Computer den Cosinus ausrechnen lassen. Das Argument liefern Sie in Bogengraden.

Beispiel: Die Anweisung PRINT COS (−0.5) bringt richtig den Wert 0.87758256 auf den Bildschirm.

DATA — daraus können Sie lesen

Hinter dieser Anweisung können Sie, durch Komma getrennt, Ihre Daten unterbringen. Das können Zahlen oder Zeichenfolgen in Anführungszeichen oder auch ohne sein. Haben Sie mehr Daten unterzubringen als in eine Zeile passen, schreiben Sie einfach

mehrere DATA-Anweisungen hintereinander, der Computer liest sie der Reihe nach.
Beispiel: DATA 1, 2, 5, "HALLO", 9870, MIST

DEF — schreiben Sie eigene Funktionen

Auf DEF folgt FN und dann der Name einer Variablen. Dahinter müssen Sie in Klammern eine numerische Variable und dann ein Gleichheitszeichen angeben. Auf das Gleichheitszeichen folgt die eigentliche Rechnung.

. . . für immer wieder auftauchende Probleme



Wenn Sie dann die Funktion aufrufen, wird die Rechnung für die mit dem Aufruf übergebene Zahl ausgeführt.
Ein einfaches Beispiel: DEF FNA (L) = 12*L. Mit dieser Anweisung definieren Sie eine Funktion unter dem Namen FNA. Der übergebene Wert wird in der Variablen L gespeichert. Er wird dann mit 12 multipliziert und das Ergebnis wird zurückgegeben.
Aufrufen können Sie die Funktion mit LET A = FNA (A). A wird dann mit 12 multipliziert. Natürlich lohnt sich das nur bei komplizierten Formeln.

DIM — Reservieren Sie Platz im Speicher

Mit der Anweisung DIM reservieren Sie im Speicher Platz für numerische Felder oder für Felder von Zeichenfolgen. Das ganze Feld wird mit 0 oder bei Zeichenfolgen mit der lee-

ren Zeichenfolge "" initialisiert. Sie können dem Feld mehrere Dimensionen mitgeben. Jede Größenangabe ist von der nächsten durch ein Komma getrennt.

Beispiel: Die Anweisung DIM A (10, 10, 10) reserviert Platz für ein dreidimensionales Feld mit dem Namen A. Insgesamt passen 1000 Werte in dieses Feld. Genauso können Sie Felder für Zeichenfolgen reservieren.

END — setzen Sie dem Programm ein Ende

Stößt der Computer bei Ausführung des Programms auf die Anweisung END, macht er Feierabend, so als hätte er die letzte Zeile des Programms ausgeführt.

EXP — rechnen Sie logarithmisch

Das Gegenstück zu EXP ist die Funktion LN. Mit diesen beiden Funktionen können Sie logarithmisch rechnen.

Beispiel: Geben Sie der Funktion EXP das Argument 1, also EXP (1), so erhalten Sie die Basis der natürlichen Logarithmen e mit dem Wert 2.712818828....

FOR — bitte wiederholen Sie

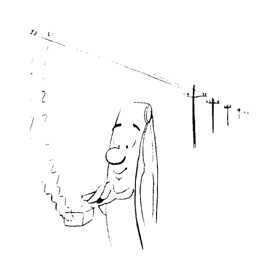
Die Schlüsselworte FOR, TO, STEP und NEXT gehören eigentlich zusammen. Sie bilden eine Einheit und stellen zusammen eine der wichtigsten Befehlsstrukturen von Basic dar.

Mit diesen Schlüsselworten können Sie den Computer anweisen, eine Anzahl von Befehlen mehrfach zu wiederholen.

Diese Wiederholungsanweisung beginnt immer mit dem Schlüsselwort FOR. Es folgt eine Variable, die festhält, in welchem Durchlauf sich die Wiederholung befindet. Dieser Variablen gibt man mit einem Gleichheitszeichen einen Anfangswert. Dann kommt das Schlüsselwort TO und danach der Endwert für die Wiederholung.

Beispiel: FOR I = 1 TO 10. Die Anweisung beginnt mit FOR, die Variable heißt I und erhält den Anfangswert 1. Nach dem Schlüsselwort TO kommt der Endwert 10.

... was soll ich wiederholen?



Damit ist die Anweisung für die Wiederholung aber noch nicht komplett. Es folgen jetzt die Befehle, die wiederholt werden sollen. Nach diesen Befehlen kommt zum Abschluß die Anweisung NEXT, gefolgt von der Variablen, die in der FOR-Anweisung festgelegt wurde. In unserem Beispiel ist das I, die Anweisung heißt also NEXT I.

Der Programmablauf ist dann so: Zuerst bekommt I den Wert 1. Dann werden alle Anweisungen bis zu NEXT I ausgeführt. Da bekommt I den Wert 2 und die Anweisungen nach dem FOR werden wieder ausgeführt, bis zum NEXT I. Dort erhält I den Wert 3 und so geht das weiter, bis I einen Wert erhält, der größer ist als der Endwert, der in unserem Fall ja 10 ist.

Wenn I also den Wert 11 erhält, bricht der Computer die Wiederholung ab und führt das Programm mit der nächsten Anweisung nach NEXT I weiter aus.

In unserem Beispiel wird der Wert von I bei der Anweisung NEXT I immer um 1 größer. Das kann man mit dem Schlüsselwort STEP ändern, das einfach noch hinter den Endwert gesetzt wird. STEP heißt Schrittweite. Hinter STEP können Sie dann die Zahl angeben, die jeweils bei der Anweisung NEXT I zur Variablen I addiert werden soll. Komplett hätte unser Beispiel also heißen müssen FOR I = 1 TO 10 STEP 1.

. . . steppen Sie einen Schritt weiter

Wenn hinter STEP aber eine 1 steht kann man STEP und die 1 auch einfach weglassen.

Wiederholungen mit FOR nennt man übrigens auch Schleife. Die Variable kann natürlich auch anders heißen als I, und der Anfangswert muß auch nicht immer 1 sein.

FRE — wieviel Platz ist eigentlich noch da?

Damit können Sie den restlichen Platz im Speicher abfragen. Hinter FRE müssen Sie in Klammern eine beliebige Zahl angeben.

Beispiel: PRINT FRE (0) gibt auf dem Bildschirm die Anzahl noch freier Speicherplätze (Bytes) aus.

GET — Abhorchen der Tastatur

Mit dem Befehl GET können Sie die Tastatur abhorchen. Wenn Sie keine Taste gedrückt haben, liefert GET als Ergebnis die leere Zeichenfolge "" oder den Wert 0. Haben

Sie aber eine Taste gedrückt, erhalten Sie als Ergebnis das Zeichen der Taste. Haben Sie also etwa die Taste A gedrückt liefert GET als Ergebnis "A". GET wartet nicht wie INPUT auf einen Tastendruck. Wenn Sie gerade keine Taste gedrückt haben, Pech. Da müssen Sie selbst in Ihrem Programm für das Warten sorgen.



Beispiel: GET A$ liefert so lange in A$ die leere Zeichenfolge "" bis Sie eine Taste drükken. Dann steht das Zeichen auf der Taste in A$. Nach diesem Befehl müssen Sie also immer den Inhalt von A$ prüfen. Wenn nichts drin steht, wurde keine Taste gedrückt.

. . . bei anderen Leuten Zeichen einkaufen

Den Befehl gibt es auch mit einem nachfolgenden Nummernzeichen (#). Er bezieht sich dann auf ein Gerät, mit dem über OPEN eine Verbindung hergestellt wurde. Hinter dem Nummernzeichen folgt die Nummer des Gerätes und dann —durch Komma getrennt— die einzulesende Variable.

Beispiel: GET #1, X$ fordert von dem unter der Nummer 1 verbundenen Gerät ein Zeichen an.

GOSUB — Sprung mit Rücksprung

Eng zusammen gehören die Befehle GOSUB und RETURN. Beide sollten Sie sich gut merken, denn mit diesen Befehlen können Sie gut strukturierte Programme schreiben.

Aus anderen Programmiersprachen kennt man sogenannte Unterprogramme. So was Ähnliches ist in Basic mit GOSUB und RETURN möglich.

Benutzt werden diese Befehle besonders dann, wenn man Teile eines Programms immer wieder ausführen will.

. . . Mehrfachnutzung möglich

Diese Anweisungen schreibt man dann hintereinander und als letzte fügt man ein RETURN hinzu. Eine solche Anweisungsfolge kann man jetzt von jeder beliebigen Stelle des Programms aus aufrufen. Man benutzt dazu den Befehl GOSUB und gibt die erste Zeilennummer der Anweisungsfolge an.



Der Computer führt jetzt diese Anweisungen aus, bis er auf RETURN trifft. Das heißt ZURÜCK. Er springt dann zurück auf die Anweisung die auf die GOSUB-Anweisung folgt.

Natürlich kann man die Anweisungsfolge von verschiedenen Stellen des Programms aus aufrufen. Der Computer kehrt dann beim RETURN selbstverständlich auch an verschiedene Stellen zurück.

. . . Nummernsalat heißt dieses Spiel

Beispiel: Nehmen wir an, die Anweisungsfolge beginnt bei der Anweisung mit der Nummer 1000. Das zugehörige RETURN hat die Nummer 1010. Wenn man jetzt unter Nummer 20 die Anweisung GOSUB 1000 schreibt, führt der Computer die Anweisungen von 1000 bis 1010 aus und springt dann auf die nächste Anweisung nach Nummer 20.

Steht dann beispielsweise unter Nummer 100 noch einmal GOSUB 1000, führt er wieder die Anweisungen von 1000 bis 1010 aus und springt dann auf die nächste Anweisung nach 100.

GOTO — Änderung der Reihenfolge

Normalerweise führt der Computer die Befehle eines Basic-Programms in der Reihenfolge ihrer Zeilennummern aus. Diese Reihenfolge kann man aber mit dem GOTO-Befehl ändern. Hinter dem GOTO kann man die Zeilennummer des Befehls angeben, der als nächster ausgeführt werden soll. Der Computer springt dann zu diesem Befehl, und dann geht alles wieder seinen gewohnten Gang.

. . . gehen Sie sofort dorthin

GOTO kann man einmal benutzen, um den Computer unwiderruflich zu einer bestimmten Anweisung zu schicken. Häufig verwendet man GOTO aber auch in einer IF-Anweisung. Der Sprung ist dann an eine Bedingung geknüpft.

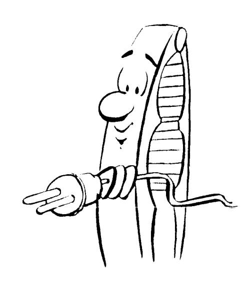
Beispiel: GOTO 1000 befiehlt dem Computer, als nächstes die Anweisung mit der Nummer 1000 auszuführen.

IF — wenn das Wörtchen wenn nicht wär'

Mit dem Befehl IF stellen Sie dem Computer eine Bedingung. Zu IF gehört das Schlüsselwort THEN. Damit sagen Sie dem Computer, was er tun soll, wenn die Bedingung erfüllt ist. IF und THEN gehören also immer zusammen.

Beispiel: Die Anweisung IF A=B THEN PRINT A, wird immer dann den Wert von A auf den Bildschirm bringen, wenn der Wert von A gleich dem Wert von B ist. Die Bedingung, die auf das Schlüsselwort IF folgt, ist also A = B. Wenn diese Bedingung erfüllt ist, soll der Computer das tun, was auf das Schlüsselwort THEN folgt, nämlich PRINT A. Ist die Bedingung aber nicht erfüllt, soll er PRINT A vergessen und gleich die folgende Anweisung ausführen.

INPUT — Eingaben erbeten



Der Befehl INPUT fordert eine Eingabe von der Tastatur an. Die Programmausführung wird so lange unterbrochen, bis der geforderte Wert eingegeben ist. Als Eingabe können Sie numerische Werte oder Zeichenfolgen anfordern, je nachdem, was für eine Variable Sie hinter INPUT angeben.

Beispiel: INPUT A fordert einen numerischen Wert an, der nach der Eingabe in A gespeichert ist. INPUT A$ fordert eine Zeichenfolge an, die nach der Eingabe über die Variable A$ verfügbar ist.

. . . darf ich Sie auffordern?

Zusätzlich können Sie in der INPUT-Anweisung auch eine Aufforderung oder einen anderen Text unterbringen, der auf dem Bildschirm erscheint. Dazu geben Sie den Text in Anführungszeichen unmittelbar hinter INPUT an. Darauf folgt ein Semikolon und danach geht es normal weiter.

Beispiel: INPUT "BITTE GEBEN SIE

JETZT IHR ALTER EIN"; A. Auf dem Bildschirm erscheint dann der Text als Aufforderung für die Eingabe. Anschließend erhält die Variable A den Wert, den Sie über die Tastatur eingeben.

. . . auf die Nummer kommt es an

Den Befehl INPUT gibt es auch mit einem nachfolgenden Nummernzeichen (#). Er bezieht sich dann auf ein Gerät, mit dem über OPEN eine Verbindung hergestellt wurde. Hinter dem Nummernzeichen folgt die Nummer des Gerätes und dann —durch ein Komma getrennt— die einzulesenden Variablen.

Beispiel: INPUT #1, X$ fordert von dem unter der Nummer 1 verbundenen Gerät eine Zeichenfolge an. Natürlich können Sie keinen Aufforderungstext angeben.

INT — einfach abgeschnitten

Die Funktion schneidet bei einer Kommazahl einfach alles ab, was hinter dem Komma steht.

Beispiel: PRINT INT(3.234215) bringt auf den Bildschirm einfach eine 3. Genauso die Anweisung PRINT INT(3). PRINT INT(A) bringt beispielsweise eine 0 auf den Bildschirm, wenn A den Wert 0.087639 hat.

LEN — die Anzahl der Zeichen

Oft ist es wichtig, wieviel Zeichen in einer Zeichenfolge (string) sind. Diesen Wert liefert

die Funktion LEN. Als Argument bekommt LEN die Zeichenkette mit.

Beispiel: PRINT LEN ("Alle meine Entchen") liefert 18. PRINT LEN (A$) liefert die aktuelle Länge der unter A$ gespeicherten Zeichenfolge. Bei der leeren Zeichenfolge "" liefert LEN als Ergebnis 0.

LET — merken Sie sich das gefälligst!

Mit der Anweisung LET weisen Sie den Computer an, im Speicher Platz zu reservieren, diesem Platz einen Namen zu geben und sich unter diesem Namen einen Wert zu merken. Platz können Sie für Zahlen oder für Zeichenfolgen reservieren.

... Platz schaffen und ablegen



Da sich der Wert unter diesem Namen ändern kann, spricht man auch bei dem Namen von einer Variablen; der Wert ist eben variabel.

Beispiel: Die Anweisung LET A = 5 reserviert Speicher für eine Zahl, gibt dem Speicher den Namen A (A ist die Variable) und legt den Wert 5 unter dem Namen A ab. A hat fortan immer den Wert 5, falls Sie den Wert nicht ändern, CLR oder NEW benutzen oder den Computer ausschalten.

... für Zahlen und Buchstaben

Genauso funktioniert's mit Zeichenfolgen. Variable, die Zeichenfolgen aufnehmen können, sind immer durch das $-Zeichen gekennzeichnet. Für eine Zeichenfolge hieße die An-

weisung also: LET A$ = "Alle meine Entchen". A$ steht also fortan für "Alle meine Entchen".

LEFT$ — bitte den linken Teil

Damit können Sie sich den Anfang einer Zeichenfolge holen. Hinter LEFT$ geben Sie in Klammern an, von welcher Zeichenfolge Sie den Anfang wollen und —durch Komma getrennt— wieviel Zeichen Sie wollen.

Beispiel: LEFT$ ("ABCDEFG", 3) liefert die Zeichenfolge "ABC".

LIST — Zeig her Dein Programm

Wenn Sie ein Programm eingetippt haben, können Sie dem Computer mit LIST den Befehl geben, das Programm auf dem Bildschirm zu zeigen. Hinter LIST können Sie noch die Nummer der Anweisung angeben, bei der der Computer beginnen soll. Wenn Sie keine Nummer angeben, beginnt der Computer mit der ersten Anweisung des Programms.

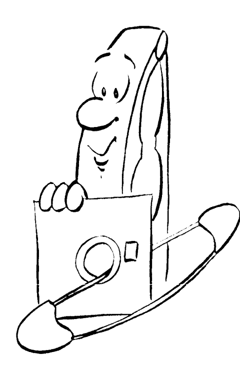
. . . ganz oder im Ausschnitt

Außerdem können Sie sich Ausschnitte des Programms auf den Bildschirm ausgeben lassen. Sie geben dazu das „Von" und das „Bis" — verbunden mit einem Bindestrich — an.

Beispiel: LIST 50 bringt das Programm ab Zeile 50 auf den Bildschirm. LIST 50—100 bringt alle Zeilen zwischen 50 und 100. LIST

bringt das ganze Programm und LIST 200 zeigt Ihnen das Programm ab Zeile 200.

LOAD — Laden Sie ein Programm



LOAD ist das Gegenstück zum Befehl SAVE. Damit können Sie ein Programm, daß Sie vorher mit SAVE auf die Kassette oder die Diskette gespeichert haben, wieder in den Computer laden. Hinter LOAD können Sie den Namen des Programms angeben, den Sie ihm beim Speichern mitgegeben haben. Der Name steht in Anführungszeichen. Wenn Sie die Anführungszeichen weglassen, lädt der Computer einfach das nächste Programm, daß er auf der Kassette oder Diskette findet. Angehängt mit einem Komma können sie die Gerätenummer angeben. Kassette (1) ist voreingestellt und kann weggelassen werden.

Beispiel: LOAD "SPIRALE" lädt das Programm von der Kassette, das sie vorher unter dem Namen "SPIRALE" gespeichert haben. Der Befehl LOAD lädt das nächste Programm von der Kassette.

LOG — für Logarithmiker

LOG ist das Gegenstück zu EXP. Wenn Sie also logarithmisch rechnen wollen, Basic bietet dazu die Möglichkeit.

MID$ — bitte ein Stück aus der Mitte

Damit können Sie sich ein Stück aus der Mitte einer Zeichenfolge holen. Hinter MID$ geben Sie in Klammern an, von welcher Zei-

chenfolge Sie das Stück wollen und —durch Komma getrennt— wo Ihr Stück anfangen soll und wieviel Zeichen Sie wollen.

Beispiel: MID$ ("ABCDEFG", 2, 3) liefert die Zeichenfolge "BCD".

NEW — alles neu macht der . . .

Der Befehl NEW hat die gleiche Wirkung wie das Aus- und Einschalten des Gerätes. Das Programm und alle gespeicherten Werte werden gelöscht. Vorsicht!!

NEXT — das Ende einer Schleife

Das Schlüsselworf NEXT gehört zur Wiederholung, die mit FOR beginnt. Unter FOR ist die Wiederholung (Schleife) ausführlich erklärt.

NOT — so geht es aber nicht

Logischer Operator "NICHT". Benutzt wird er hauptsächlich in IF-Anweisungen. Damit wird die Bedingung umgekehrt.

ON — wohin geht die Reise?

Mit ON können Sie abhängig von einem Wert zu einer bestimmten Stelle des Programms springen, ohne ganz viele IF-Anweisungen zu benutzen. Auf ON folgt ein Wert oder eine Formel. Dann GOTO oder GOSUB. Schließlich folgen —durch Komma getrennt— die Nummern von Anweisungen, zu

denen man möglicherweise springen will. Wenn der Wert oder das Ergebnis der Formel 1 ist, springt das Programm zur ersten angegebenen Anweisungsnummer, bei 2 zur zweiten usw.

Kommt kein passendes Ergebnis, wird einfach die nächste Anweisung nach ON ausgeführt.

Beispiel: ON A GOTO 100, 200, 300 bewirkt einen Sprung zu Zeile 100, wenn A = 1 ist, zu 200 bei A = 2 und zu 300 bei A = 3. In allen anderen Fällen führt der Computer die Anweisung nach der ON-Anweisung aus.

OPEN — Sesam, öffne dich

Wenn Sie irgendwas auf den Drucker, die Kassette oder die Diskette schreiben wollen, benutzen Sie den Befehl OPEN, um eine Verbindung zu schaffen. Hinter OPEN geben Sie eine Zahl an, unter der Sie in Zukunft das Gerät ansprechen wollen. Dann folgt die Nummer für das ausgesuchte Gerät, etwa eine 1 für Kassette oder 4 für den Drucker.

Beispiel: OPEN 1, 4 macht Ihnen den Drucker künftig unter der Nummer 1 verfügbar, bis Sie die Verbindung mit CLOSE unterbrechen.

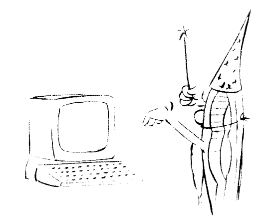
OR — das logische Oder

Die wichtigste Aufgabe erledigt das Schlüsselwort OR in der IF-Anweisung. Da stellen Sie ja dem Computer eine Bedingung. Wenn die Bedingung erfüllt ist, dann tue das und das. Nun kann es aber auch notwendig sein, die Ausführung einer Anweisung an

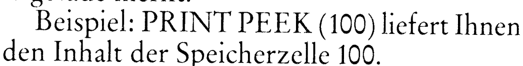
mehrere Bedingungen zu knüpfen. Sie soll nur dann ausgeführt werden, wenn eine von mehreren Bedingungen erfüllt ist. Dafür gibt es das Schlüsselwort OR.

Beispiel: IF A=10 OR A=15 THEN PRINT A. Bei dieser Anweisung bekommen Sie den Wert von A nur auf den Bildschirm, wenn er entweder 10 oder 15 ist. Der Befehl PRINT A wird also nur dann ausgeführt.

PEEK — picken Sie aus dem Speicher



Normalerweise müssen Sie sich nicht darum kümmern, wo der Computer etwas im Speicher unterbringt. Wenn Sie aber mal eine ganz bestimmte Stelle im Speicher lesen wollen, können Sie das mit PEEK. Dieses Lesen im Speicher hat natürlich nur Sinn, wenn Sie wissen, was sich der Computer an dieser Stelle gerade merkt.



Beispiel: PRINT PEEK (100) liefert Ihnen den Inhalt der Speicherzelle 100.

POKE verändern Sie den Speicher

Wenn Sie direkt einen Wert im Speicher unterbringen wollen, können Sie das mit dem Befehl POKE. Er ist das Gegenstück zu PEEK. Sie sollten diesen Befehl nur sehr vorsichtig benutzen und sich genau überlegen, wo Sie ein Zeichen im Speicher unterbringen wollen. Außerdem sollten Sie immer dafür sorgen, daß das Basic dadurch nicht „gestört" wird. Denn sonst passieren auf einmal ganz merkwürdige Sachen.

Meist benutzt man den POKE-Befehl um Maschinenprogramme im Speicher unterzubringen.

Doch das ist eine Aufgabe für Spezialisten.

Ansprechen können Sie mit dem Befehl POKE die Speicherzellen 0 bis 65535 (64 k), sofern Ihr Computer über so viel Speicher verfügt.

. . . viel Platz ist da nicht

In jeder Speicherzelle können Sie einen Wert zwischen 0 und 255 unterbringen. Mehr paßt da nicht rein. Zuerst müssen Sie die Nummer der Speicherzelle angeben und dann, durch Komma getrennt, den Wert, den Sie speichern wollen.

Beispiel: POKE 1000, 115 legt in der Speicherzelle 1000 den Wert 115 ab.

POS — welche Zeile wollen Sie?

Mit POS können Sie erfahren, in welcher Zeile die nächsten PRINT-Anweisung ausgeführt wird. Sie müssen POS in Klammern eine Zahl mitgeben, deren Wert jedoch ohne Bedeutung ist.

PRINT — so schreibt der Computer

PRINT heißt soviel wie DRUCKE. Nun kann man zwar auf dem Bildschirm nicht drucken, aber als die Sprache Basic entwickelt wurde, arbeitete man noch nicht mit Bildschirmen, sondern ausschließlich mit Drukkern.

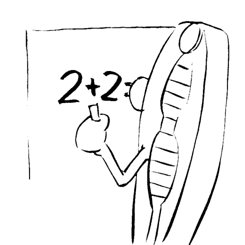
Heute geben Sie in einer PRINT-Anweisung dem Computer an, was er auf den Bildschirm schreiben soll.

Angeben können Sie Zahlen und Zeichenfolgen, sowohl direkt als auch in Form von Variablen.

Beispiel: PRINT A$ bringt auf den Bildschirm die unter A$ gespeicherte Zeichenfolge. PRINT A bringt den unter A gespeicherten Wert auf den Bildschirm. Und PRINT "Alle meine Entchen" bringt alles auf den Bildschirm, was zwischen den Anführungszeichen steht.

... Rechenaufgaben unterwegs gelöst

In der PRINT-Anweisung können Sie auch Rechenaufgaben lösen oder Funktionen benutzen.



Beispiel: PRINT 7+5 bringt auf den Bildschirm das Ergebnis dieser Aufgabe, nämlich 12. PRINT RND (0) liefert Ihnen die nächste Zufallszahl.

Die Stelle auf dem Bildschirm, wo die Werte erscheinen sollen, können Sie mit den Schlüsselworten POS, SPC und TAB bestimmen. Alle drei sind separat erklärt.

... Punkt, Punkt, Komma, Strich ...

Wichtig sind in der PRINT-Anweisung noch Komma und Semikolon. Denn Sie können auch mehrere Werte in einer PRINT-Anweisung angeben. Diese Werte müssen dann durch Komma oder Semikolon voneinander getrennt sein. Benutzen Sie das Komma, läßt der Computer automatisch Platz hinter jedem Wert. Benutzen Sie das Semikolon, folgen die Werte unmittelbar aufeinander.

Beispiel PRINT "7+5=";7+5 bringt auf den Bildschirm: 7+5=12. Hätten Sie statt des

Semikolons ein Komma benutzt, wäre auf dem Bildschirm vor der 12 viel leerer Platz.

Komma und Semikolon haben auch am Ende der PRINT-Anweisung Bedeutung. Normalerweise beginnt der Computer mit jeder neuen PRINT-Anweisung eine neue Zeile. Das unterbinden Sie, wenn Sie an den Schluß ein Komma oder ein Semikolon setzen. Beispiel: Folgen in einem Programm die beiden Anweisungen PRINT "Alle meine"; und PRINT "Entchen" aufeinander, so kommt der Anfang dieses Liedes in eine Zeile, obwohl zwei PRINT-Anweisungen benutzt wurden.

. . . schreiben auf andere Geräte



Den Befehl PRINT gibt es auch mit einem nachfolgenden Nummernzeichen (#). Er bezieht sich dann auf ein Gerät, mit dem über OPEN eine Verbindung hergestellt wurde. Hinter dem Nummernzeichen folgt die Nummer des Gerätes und dann —durch Komma getrennt— die auszugebenden Variablen.

Beispiel: PRINT #1,A$ liefert auf Gerät 1 die Zeichenfolge A$ ab. TAB und SPC „verstehen" übrigens nicht alle Geräte.

READ — lesen Sie von DATA

Mit dem Befehl READ können sie die Daten aus der DATA-Anweisung den Variablen zuweisen. Jede Variable in der READ-Anweisung bekommt einen Wert aus der DATA-Anweisung. Vorsicht, die Typen müssen übereinstimmen.

Beispiel: READ A,A$ liest aus der DATA-

Anweisung eine Zahl, die sich dann in A wiederfindet und anschließend eine Zeichenfolge, die dann in A$ abgelegt wird.

REM — kommentieren Sie heftig

Der Befehl REM kommt von remember, sich erinnern. Wenn Sie ein Programm schreiben, sollten Sie den REM-Befehl häufig gebrauchen und alles Erinnernswerte notieren. Schon wenn Sie ein Programm ein paar Tage nicht mehr benutzt haben, aber noch eine Änderung anbringen wollen, werden Sie diese Art von Erinnerung zu schätzen wissen.

. . . geben Sie sich eine Chance für später

Wenn gar ein anderer Programmierer Ihr Programm ändern will, hat er ohne diese Kommentare kaum eine Chance, besonders wenn das Programm sehr lang ist.

Natürlich verlieren Sie durch die REM-Zeilen Speicher. Doch eigentlich müßte für einen Kommentar immer Platz sein. Nach dem REM können Sie die ganze Zeile vollschreiben und Sie dann abschließen.

Beispiel: REM DIES IST EIN KOMMENTAR

RESTORE — zurück, und nochmal von vorn

RESTORE steht in enger Verbindung mit DATA und READ. READ liest ja die Werte aus den DATA-Anweisungen, einen nach dem anderen. Wenn Sie aber noch einmal von vorn beginnen wollen, geben Sie einfach den Befehl RESTORE. Das Programm beginnt

dann von vorn mit dem ersten Wert der ersten DATA-Anweisung. Damit kann man Daten mehrfach verwenden.

RETURN — springen Sie zurück

Die Bedeutung von RETURN ist unter GOSUB erklärt. RETURN läßt den Computer zu der Anweisung zurückspringen, die dem GOSUB folgt.

RIGHT$ — bitte den rechten Teil

Damit können Sie sich das Ende einer Zeichenfolge holen. Hinter RIGHT$ geben Sie in Klammern an, von welcher Zeichenfolge Sie das Ende wollen und —durch Komma getrennt— wieviel Zeichen Sie wollen.

Beispiel: RIGHT$ ("ABCDEFG",3) liefert die Zeichenfolge "EFG".

RND — das kann doch kein Zufall sein

Die Funktion RND liefert Ihnen Zufallszahlen zwischen 0 und 1. Der Funktion RND müssen Sie eine Zahl mitgeben. Die erste Zahl bestimmt die Zufallszahlen.

. . . nicht zufällige Zufälle

Allerdings sind es nicht wirklich zufällige Zahlen. Aber zumindest sind sie ziemlich durcheinander. Für alle weiteren Aufrufe von RND sollte die mitgegebene Zahl 0 oder positiv sein. Eine negative Zahl startet eine neue

Reihe von Zufallszahlen. Die müßten Sie dann wieder zufällig aussuchen.

Beispiel: PRINT RND(0) liefert Ihnen eine Zufallszahl auf den Bildschirm.

RUN — damit geht's richtig los

Mit RUN starten Sie das Programm, das gerade im Speicher ist. Hinter dem Schlüsselwort RUN können Sie auch die Nummer einer Anweisung angeben. Der Computer beginnt mit der Ausführung des Programms dann bei dieser Anweisung. Sonst fängt er mit der ersten Anweisung des Programms an. RUN löscht übrigens alle gespeicherten Werte, bevor das Programm startet.



SAVE — speichern auf Kassette

Wenn Sie ein Programm auf der Kassette oder Diskette für späteren Gebrauch speichern wollen, benutzen Sie das Kommando SAVE.

. . . einen Namen sollten Sie immer vergeben

Hinter SAVE können Sie noch den Namen angeben, unter dem Sie das Programm später wiederfinden wollen, außerdem die Nummer des Gerätes. Gerät 1 (Kassette) ist voreingestellt und kann weggelassen werden.

Um das Programm wieder von der Kassette oder Diskette zu lesen, benutzen Sie das Kommando LOAD.

Beispiel: SAVE "EIMER" sichert das im

Speicher befindliche Programm unter den Namen EIMER auf die Kassette. SAVE "EIMER",8 speichert das Programm auf Gerät 8 (Diskette).

SGN — auf das Vorzeichen kommt es an

Wenn Sie wissen wollen, ob eine Zahl positiv, negativ oder 0 ist, benutzen Sie die Funktion SGN. Sie liefert —1 für alle negativen und +1 für alle positiven Zahlen. Bei einer 0 kommt auch eine 0 zurück.

Beispiel: PRINT SGN(—311) liefert —1.

SPC — schaffen Sie Platz

Mit SPC können Sie eine Anzahl von Leerzeichen in der PRINT-Anweisung angeben. Hinter SPC geben Sie in Klammern die Anzahl der Zeichen an. Diese Zeichen werden dann einfach übersprungen.

Beispiel: PRINT"ABC";SPC(20);"DEF" setzt auf dem Bildschirm einen Leerraum von 20 Zeichen zwischen "ABC" und "DEF".

SIN — liefert den Sinus

Wenn Sie den Sinus einer Zahl kennenlernen wollen, hilft Ihnen die Funktion SIN. Geben Sie ihr einfach die Zahl als Argument.

SQR — das Übel an der Wurzel packen

Die Funktion SQR bietet Ihnen die Quadratwurzel.

Beispiel: PRINT SQR (64) liefert 8 und PRINT SQR (16) gibt natürlich 4.

STEP — wie groß sind Ihre Schritte?

STEP gehört zur Wiederholung, die mit FOR beginnt. Dort ist STEP ausführlich beschrieben.

STOP — stehenbleiben, oder . . .

Trifft der Computer auf diesen Befehl, unterbricht er die Ausführung des Programms. Wenn Sie ihn weiterarbeiten lassen wollen, benutzen Sie die Anweisung CONT. Der Computer nennt Ihnen auch die Zeile, in der er den Befehl gefunden hat.

STR$ — eine Zahl als Zeichenkette

Mit der Funktion STR$ können Sie eine Zahl in eine Zeichenkette umwandeln.

Beispiel: Wenn A den Wert 789 hat, können Sie mit LET A$ = STR$ (A) die Zeichenfolge "789" A$ zuweisen.

SYS — Unterprogramme im Speicher

Mit SYS können Sie Unterprogramme aufru-



fen, die im Speicher abgelegt sind. Die Programme müssen in Maschinensprache geschrieben sein.

TAB — legen Sie Positionen fest

TAB können Sie in der PRINT-Anweisung verwenden. Sie geben damit die Spalte an, in die das nächste Zeichen kommen soll.

Beispiel: PRINT TAB 10; "HALLO" läßt das Wort HALLO in Spalte 10 beginnen.

TAN — liefert den Tangens

Mit dem Tangens sind die Kreisfunktionen komplett. TAN funktioniert wie alle anderen Kreisfunktionen auch.

THEN — nur wenn die Bedingungen stimmen

Das Schlüsselwort THEN gehört zur IF-Anweisung und ist auch da erklärt. Hinter THEN geben Sie an, was passieren soll, wenn die Bedingungen hinter IF erfüllt sind.

TO — bis dahin geht die Wiederholung

Das Schlüsselwort TO gehört zur FOR-Anweisung und ist auch da erklärt. Hinter TO



geben Sie die obere Grenze für eine Wiederholung an.

VAL — Zahl aus der Zeichenfolge

Wenn Sie eine Zahl als Zeichenfolge angeben, liefert Ihnen VAL den korrekten Wert.

Beispiel: PRINT VAL ("723") liefert die Zahl 723.

VERIFY — wie eine Versicherung

Wenn Sie ein Band auf Kassette oder Diskette gesichert haben, wollen Sie ja immer ganz sicher sein, daß es auch richtig angekommen ist. Die Überprüfung erledigt VERIFY für Sie. Der Computer vergleicht dann das Programm im Speicher mit dem auf der Kassette oder der Diskette.

Geben Sie einfach nur VERIFY an, so sucht sich der Computer das nächste Programm auf der Kassette. Sie können aber auch einen Namen mitliefern. Dann sucht der Computer zuerst das Programm.

Beispiel: VERIFY "FRITZ" sucht das unter "FRITZ" gespeicherte Programm und vergleicht es mit dem Programm im Speicher.

WAIT — Lassen Sie den Computer warten

Mit dem Befehl WAIT können Sie ein Programm so lange warten lassen, bis eine bestimmte Speicherzelle einen bestimmten Wert angenommen hat.

Die neue Mikro- und Home-Computer-Reihe „Nußknacker"!



Wir knacken Ihre Programmnüsse!
Gut dokumentierte Programme, die wirklich funktionieren.

Die ersten 5 Bände:

Neue Spiele und Programme zu ZX81 und Spectrum

Hrsg. Steffi Siol. 1984. 128 Seiten mit 70 Abbildungen, Format 13 x 21 cm, kartoniert, ca. DM 29,80. ISBN 3-481-36311-7
Dieses Buch liefert „jede Menge Futter" für „Spielratten" und Hobbyprogrammierer, die mit dem populären ZX81 oder dem Spectrum „computern".

Hilfe, Textverarbeitung mit dem Computer
Eine heiter ernste Einführung

Hrsg. Steffi Siol. 1984. 96 Seiten mit 40 Abbildungen, Format 13 x 21 cm, kartoniert ca. DM 28,–. ISBN 3-481-36351-6
Diese Einführung in die Textverarbeitung entstand aus ca. 300 Interviews mit „Büroarbeitern", die auf Textautomaten oder Computer umgestiegen sind. Die wirklichen Probleme – heiter aufbereitet – bestimmen daher den Inhalt des Buches, nicht die Versprechungen der Hersteller und auch nicht die unverständlichen technischen Gebrauchsanleitungen.

Von Computern habe ich eigentlich keine Ahnung
Eine ungewöhnliche Einführung in die Welt der Heimcomputer

Hrsg. Steffi Siol. 1984. 96 Seiten mit 48 Abbildungen, Format 13 x 21 cm, kartoniert ca. DM 28,–. ISBN 3-481-36341-9
Die Autoren haben mehr als 500 Kinder, Jugendliche und Erwachsene interviewt. „Welche Fragen haben Sie zum Thema Heimcomputer, was interessiert Sie in diesem Bereich, wo wünschen Sie sich mehr Information?"

Druckerspiele und ernst Gedrucktes
Aktuelle Nußknacker-Programme

Hrsg. Steffi Siol. 1984. 112 Seiten mit 60 Abbildungen, Format 13 x 21 cm, kartoniert ca. DM 32,80. ISBN 3-481-36331-1
Endlich sind die Ergebnisse greifbar, nicht flüchtig auf dem Bildschirm, nein, schwarz auf weiß oder gar bunt gedruckt auf Papier. Bilder, Texte, Briefe, Tabellen, Etiketten, Graphiken, Figuren und was sich sonst noch drucken läßt. Auch als „Spielzeug" ist der Drucker geeignet. Einige lustige Spielprogramme in diesem Buch beweisen das.

Neue Spiele und Programme zum Commodore 64

Hrsg. Steffi Siol. 1984. 128 Seiten mit 60 Abbildungen, Format 13 x 21 cm, kartoniert ca. DM 32,80. ISBN 3-481-36321-4
Dieses Buch liefert „jede Menge Futter" für „Spielratten" und Hobbyprogrammierer, die mit dem Commodore 64 „computern".

Weitere Bände in Vorbereitung.



Verlagsgesellschaft
Rudolf Müller GmbH
Postfach 41 09 49
5000 Köln 41



Das Nußknacker-Buchteam bietet Ihnen als besonderen Service individuelle Beratung zum Thema:

COMMODORE 64

Schneiden Sie diesen Gutschein aus, schreiben Sie Ihre Frage darauf, vergessen Sie Ihren Absender nicht und schicken den Gutschein zusammen mit drei DM in Briefmarken an umseitige Anschrift.

Ihre Frage:

____

____

____

____

____

Ihre Anschrift:

____

____

____

____



Falten Sie diesen Gutschein in der Mitte. Er paßt dann in einen Fensterumschlag.

(Hier knicken)

Verlag Rudolf Müller
Nußknacker-Buchteam
Stichwort „Frage C64“
Postfach 41 09 49

5000 Köln 41

Programmierer

Wir sind ständig an neuen Programmen interessiert. Senden Sie uns Ihre Programme mit vollständiger Dokumentation. Wir prüfen Programm, Anleitung und Kommentierung. Sie erhalten ein Honorar, wenn Ihr Programm in unserer Buchreihe veröffentlicht wird.

Legen Sie diese Seite Ihrem Programm ausgefüllt bei. Unsere Anschrift steht auf der Rückseite dieses Blattes.

Name: ______________________________

Anschrift: ______________________________

____

Telefon: ______________________________

Programm für Rechner: ______________________________

____

Hardware-Voraussetzungen: ______________________________

____

Software-Voraussetzungen: ______________________________

____

Honorarvorstellung: ______________________________



Falten Sie diesen Gutschein in der Mitte. Er paßt dann in einen Fensterumschlag.

– – (Hier knicken) –

Verlag Rudolf Müller
Nußknacker-Buchteam
Stichwort „Programm“
Postfach 41 09 49

5000 Köln 41



Wir knacken Ihre Programmnüsse! Gut dokumentierte Programme, die wirklich funktionieren

Aus dem Inhalt

Dieses Buch liefert „jede Menge Futter" für „Spielratten" und Hobbyprogrammierer, die mit dem Commodore 64 „computern". Die Programmierung bleibt kein Buch mit sieben Siegeln. Viele Anregungen und Hinweise auf mögliche Erweiterungen der Programme fordern eigene Phantasie und Programmierkünste und natürlich „klappt" alles richtig Eingetippte.



M Verlag Rudolf Müller
Köln

ISBN 3-481-36321-4